Tage Røsten

Seismic data compression using subband coding

NTNU Trondheim
Norges teknisk-naturvitenskapelige
universitet

Doktor ingeniøravhandling 2000:61
Institutt for teleteknikk
420004

Tage Røsten

Seismic data compression using subband coding

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doktor Ingeniør

Department of Telecommunications
Norwegian University of Science and Technology
N-7491 Trondheim
Norway

August 2000

"Can't you see
It all makes perfect sense
Expressed in dollars and cents
Pounds shillings and pence
Can't you see
It all makes perfect sense"

Roger Waters

ISBN 82-7984-086-9
ISSN 0809-103X

Abstract

Interpreters of seismic data struggle with the limitations of their hardware and software every day. The archiving and transporting of seismic data volumes require huge transmission capacities and large storage mediums. A large seismic survey typically comes to more than one terabyte; hence the seismic industry must search for effective ways of handling data. To work around these problems, geoscientists have traditionally decreased the size of the seismic data volume for their interpretation. Examples include reduction of the temporal range and/or resampling in time, reduction of the spatial range and/or resampling in space, and clipping of the amplitudes, thereby reducing the dynamic range.

Different from conventional or conservative approaches, lossy compression using subband coding is in this dissertation considered as the most desirable way of improving the efficiency of transmission and storage of seismic data. Depending upon the required fidelity of the reconstructed (decompressed) seismic data, the goal is to obtain compression ratios of 10:1-20:1 for prestack seismic data. The ability to compress seismic data for example 10:1 means that 10 times more data can be stored in the space that it takes to store the original data set. Compression of 10:1 may for instance help to surmount the ever-recurring problem of shortage of disk space for processing, interpretation, and permanent storage.

The seismic industry is careful not to use procedures that introduce noise into the original seismic data set. Nevertheless, speaking about digital seismic data without noise is meaningless due to the sampling of a non-band-limited seismic signal in time and space with subsequently quantization of the amplitudes. Lossy seismic data compression can be seen as just one out of several minor noise sources in the acquisition and processing of seismic data. In best case, seismic data quality can even be increased by correct use of lossy compression since ambient noise components at high frequencies and high wavenumbers are effectively reduced.

With an understanding of the associated drawbacks it is expected that the need for reduced turnaround times, from seismic data acquisition to processing and interpretation, will push the use of seismic data compression as a routine component of standard survey operations.

The dissertation is composed of three (submitted journal) papers:

- Seismic data compression,
  Part I: Subband coding of common offset gathers;

- Seismic data compression,
  Part II: Lossless coding of trace identification headers;

- Seismic data compression,
  Part III: Its influence on processing and interpretation.

The first paper deals with lossy compression of prestack seismic data using subband coding, the second paper is about lossless coding of the associated trace headers using entropy coding, and the third paper analyzes the influence of lossy seismic data compression on processing and interpretation.

A copy of the individual abstracts to the papers follows:

Seismic data compression,
Part I: Subband coding of common offset gathers

Seismic data require huge transmission capacities and large storage mediums, and the volume of seismic data is continuing to increase rapidly due to, e.g., acquisition of large 3-D surveys, re-processing of prestack seismic data, and calculation of poststack seismic data attributes. We consider lossy compression as an important tool for efficient handling of large seismic data sets. We present a 2-D lossy seismic data compression algorithm based on subband coding, and focus on adaptation and optimization of the method for common offset gathers. The subband coding algorithm consists of five stages: First, a preprocessing phase using an automatic gain control to decrease the non-stationary behavior of seismic data. Second, a decorrelation stage using a uniform analysis filter bank to concentrate the energy of seismic data into a minimum number of subbands. Then, an iterative classification algorithm based on the variance to blocks of subband samples to classify the subband samples into a fixed number of classes with approximately the same statistics. Fourth, a quantization step using a uniform scalar quantizer which gives an approximation of the subband samples to allow for high compression ratios. And finally, an entropy coding stage using a fixed number of arithmetic encoders matched to the corresponding statistics of the classified and quantized subband samples to actually achieve compression. Decompression basically performs the opposite operations in reverse order. We compare the proposed subband coding algorithm with three other seismic data compression algorithms. The high performance of our subband coding scheme is supported by objective and subjective results. For the common offset gather example, a compression ratio of 10:1 gives compression noise below the level of the ambient noise, while the compression noise is hardly visible at a compression ratio of 30:1.

Seismic data compression, Part II: Lossless coding of trace identification headers

Trace headers are an integral part of seismic data. Uncompressed trace headers become a significant portion of the total volume of seismic data in case the associated trace data samples are compressed. We present an efficient lossless compression algorithm applicable for the trace headers of the SEG-Y standard. The proposed technique uses differential coding and run-length coding in conjunction with class-wise entropy coding of the symbols (run) and the counts (length). Four entropy coding methods are compared: non-conditional semi-adaptive Huffman coding, non-conditional adaptive Huffman coding, non-conditional semi-adaptive arithmetic coding, and finally, conditional adaptive arithmetic coding. Typical compression ratios, in the case of trace headers of the common offset gather examples, are about 25:1 for the three non-conditional entropy coding methods and around 250:1 for the conditional entropy coding method. For low and medium orders, the compression ratio increases monotonically as a function of the order of the conditional probability model. The best suited order is 8. After compression, the size of trace headers becomes, independently of the applied entropy coding method, negligible compared to the size of the associated trace data samples.

Seismic data compression, Part III: Its influence on processing and interpretation

We investigate the impact of lossy seismic data compression on processing and interpretation of a 2-D seismic data set acquired in the North Sea. The seismic data set is sorted into common offset gathers which are separately compressed using a 2-D subband coding method. The associated trace headers are compressed by a lossless arithmetic coding technique. After compression and decompression, we study the influence of the introduced coding noise to a representative processing sequence. Important processing steps, in chronological order, include attenuation of water-bottom multiples, true amplitude recovery, predictive deconvolution, attenuation of peg-leg multiples, migration, and stacking. In general, migration and stacking reduce the effect of the coding noise at all compression levels, while for high compression ratios (i.e., much greater than 10:1) the coding noise has a destructive effect on the predictive deconvolution step and on the two applied multiple attenuation methods. Two sequences of the lossy compression method are explored. In the first sequence the seismic data are compressed *before* predictive deconvolution (denoted pre-decon compression), while in the second sequence the seismic data are compressed *after* predictive deconvolution (denoted post-decon compression). Several seismic data responses are examined, for example prestack and poststack amplitude analysis in addition to poststack inversion. For pre-decon and post-decon compression, compression ratios between 7.5:1-15:1 and 15:1-30:1, respectively, provide excellent reconstruction quality of the seismic data set.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doktor Ingeniør at the Department of Telecommunications, the Norwegian University of Science and Technology. Professor Tor A. Ramstad at the Department of Telecommunications, the Norwegian University of Science and Technology, and Dr. Lasse Amundsen at the Statoil Research Centre have been supervisors.

The main topic addressed is lossy compression of seismic data using subband coding, the focus being optimization of the method with an analysis of the effects of coding noise on processing and interpretation. All the three main chapters (i.e., from Chapter 2 to Chapter 4) of this thesis have been submitted to Geophysics, an international journal published by the Society of Exploration Geophysicists. There have, however, been modifications and additions of the submitted material based on comments from the Associate Editor of the journal and one reviewer (except Chapter 4), in addition to in-house corrections.

The work, including the compulsory courses corresponding to full-time studies in two semesters as well as one year of teaching assistant duties, has taken place in the period from April 1996 to June 2000. The research was carried out at the Department of Telecommunications, the Norwegian University of Science and Technology, and at the Statoil Research Centre, both located in Trondheim, Norway.

The work was funded by scholarships from Statoil (37.5 %), Petroleum Geo-Services (37.5 %), and the Department of Telecommunications, the Norwegian University of Science and Technology (25 %).

Acknowledgments

To all people who contributed, directly or indirectly, to the realization of this thesis I am indebted. In particular, I thank my supervisors Professor Tor A. Ramstad at the Department of Telecommunications, the Norwegian University of Science and Technology, and Dr. Lasse Amundsen at the Statoil Research Centre.

I highly appreciate Laurent C. Duval, Institut Français du Pétrole, for being my co-author on a conference paper and for the hundreds of electronic mails, passing both ways between France and Norway, discussing common research problems.

In addition, I am grateful to former and present colleagues at the Department of Telecommunications, the Norwegian University of Science and Technology, for all cooperation, help and support. I would particularly mention Patrick Waldemar, Are Hjørungnes, Ilangko Balasingham, and Helge Coward.

I am also greatly obliged to Sven O. Aase and John M. Lervik, two former research fellows at the Department of Telecommunications, the Norwegian Institute of Technology, for providing excellent subband image coding procedures which made the basis of my own research.

Several graduate students, whom I have been advising, have made contributions to this dissertation. In chronological order they are: Jørgen Tegdan, Tore Fevang, and Viktor H. Marthinussen.

I would also like to acknowledge the contributions of, and clarifying discussions with, colleagues at the Statoil Research Centre; Arild Buland, Andy Morton, Rune Allnor, Are Osen, Anders Sollid, Norman Ettrich, Kenneth Duffaut, Egil Holvik, Mark Thompson, and Åge Kristensen.

I thank Dr. Peter Vermeer, Associate Professor Jan E. Ødegård, and Associate Professor Geir Øien for accepting to form the evaluating committee.

The financial support of Statoil and Petroleum-Geo Services is highly acknowledged. In addition, I am obligated to Statoil for providing its computer facilities and for donating the 2-D seismic data set.

I would like to send special thanks to my parents for their support. My deepest gratitude is for my wife Toril and our sons Stian and Martin. I am deeply thankful to Toril for her kind patience and unwaring confidence in me. The many hours with Stian building Lego and Brio, reading Donald Duck and Pondus, watching Kaptein Sabeltann, Buster and Gløggen, and playing football have contributed tremendously by keeping my concentration away from science and technology when most needed.

Trondheim, August 2000

Tage Røsten

Contents

Illustrations

Tables

Nomenclature

Roman, greek, and calligraphic letters

a	coarse approximation or lowpass filtered signal
AC	non-conditional semi-adaptive arithmetic encoder operator
AC^{-1}	non-conditional semi-adaptive arithmetic decoder operator
${}^{\mathcal{O}}$AC	conditional adaptive arithmetic encoder operator of order $\mathcal{O}$
${}^{\mathcal{O}}$AC^{-1}	conditional adaptive arithmetic decoder operator of order $\mathcal{O}$
AGC	automatic gain control operator
AGC^{-1}	inverse automatic gain control operator
α	sampling coefficient
$\mathcal{A}$	distinct value or symbol to be compressed in entropy coding
b	byte number in the trace identification header
$B \times B$	block size (height × width) of input samples
$\tilde{B} \times \tilde{B}$	block size (height × width) of subband samples
β	dead-zone or threshold parameter of scalar quantizer
$\mathcal{BL}$	"Base Lysing" reference horizon used in amplitude analysis
c	class number
C	number of classes
$\mathcal{C}$	number of counts of specific symbol in run-length coding
d	detail or highpass filtered signal
d_c	proportionality constant of class number c
DC	differential encoder operator
DC^{-1}	differential decoder operator
DPCM	differential pulse code modulation operator
DPCM^{-1}	inverse differential pulse code modulation operator
δ	time-power constant
Δ	step-size of scalar quantizer
EC	entropy encoder operator
EC^{-1}	entropy decoder operator
ϵ	stop-criterion used in classification

ε	error function used in filter bank optimization
f	frequency in Hz
F	fold or multiplicity of common mid-point gathers
FB	analysis filter bank operator
FB^{-1}	synthesis filter bank operator
g	receiver coordinate
g_m	synthesis filter in the mth channel
G_{SBC}	coding gain to subband coding over pulse code modulation
$\tilde{G}_{\mathrm{SBC}}$	maximum theoretical coding gain
γ	quantizer representation level
h	offset coordinate or receiver number
h_m	analysis filter in the mth channel
H	Shannon or first-order entropy
i	quantizer index
I	number of quantizer representation levels
j	a) summation index
	b) trace number
J	number of traces
k	a) subband sample number
	b) index of letter in message
K	a) number of subband samples in one subband
	b) length of message or number of letters in message
$\mathcal{K}(\cdot)$	number of occurrences of a particular symbol in a message
l	filter tap number
L	number of analysis and synthesis filter taps
$\bar{L}$	average length of code word
m	a) subband channel number
	b) symbol index of distinct symbols
M	a) number of subbands
	b) total number of distinct symbols
n	(discrete) time sample index
n_0	delay introduced by a filter bank
N	number of input samples
$\mathcal{N}(0, \sigma^2)$	zero-mean Gaussian distribution with variance equal to σ^2
$\mathcal{O}$	order of conditional probability model
$p_X(x)$	probability density function of a stochastic variable $X \in \mathbb{R}$
$[P_b, P_e)$	probability range in arithmetic coding
$P_X(x)$	non-conditional probability model of X
$P_X^{\mathcal{O}}(x)$	conditional probability model of order $\mathcal{O}$ of X

q	output of scalar quantizer
Q	stochastic variable for the sequence of quantizer indices
r_{xx}	autocorrelation function of the input signal
R_0	intercept stack section
R_1	gradient stack section
$R(\Theta)$	reflection coefficient as a function of incident angle
$R:1$	compression ratio of trace data samples
$\tilde{R}:1$	compression ratio of trace headers
RLC	run-length encoder operator
RLC^{-1}	run-length decoder operator
ρ_1	normalized correlation coefficient at lag one
ρ_2	normalized correlation coefficient at lag two
s	a) shot coordinate or shot number b) message or input letter in entropy coding
S	a) number of levels in tree-structured filter banks b) stochastic variable for the sequence of input letters
SQ	scalar quantizer operator
SQ^{-1}	inverse scalar quantizer operator
$\hat{\sigma}_y^2$	estimated block variance of subband samples
$\tilde{\sigma}_d^2$	variance decision level in classification
$\tilde{\sigma}_r^2$	variance representation level in classification
Σ^2	stochastic variable for estimated block variances
$\mathcal{S}$	symbol in run-length coding
t	time in s
T	total width of dead-zone around zero of scalar quantizer
Θ	angle of incidence
u	common mid-point coordinate or common mid-point number
v_{rms}	(root-mean-square or) stacking velocity
w	weight factor in filter bank optimization
x	input signal
$\hat{x}$	reconstructed signal
X	stochastic variable for input signal
y_m	subband samples in the mth channel
$\hat{y}_m$	reconstructed subband samples in the mth channel
Y	stochastic variable for subband samples

Symbols

$\forall$	for all
$\lfloor x \rfloor$	the greatest integer $\leq x$
$\lceil x \rceil$	the smallest integer $\geq x$
$\downarrow M$	down-sampling by M
$\uparrow M$	up-sampling by M
$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{Z}$	the set of integers including zero

Miscellaneous

▽	receiver group
◍	shot
●	trace
Ѧ	well (or borehole)
(Σ)	summation symbol used in illustrations

Abbreviations

1-D	one-dimensional
2-D	two-dimensional
3-D	three-dimensional
4-D	four-dimensional
AC	arithmetic coding
ACF	autocorrelation function
AGC	automatic gain control
AI	acoustic impedance
AR	autoregressive
AR(1)	first-order autoregressive
AR(2)	second-order autoregressive
AVO	amplitude variation with offset
CDP	common depth-point
CMP	common mid-point
COG	common offset gather
CPU	central processing unit
CR	compression ratio
CRG	common receiver gather
CSG	common shot gather
DC	differential coding
DCT	discrete cosine transform
DPCM	differential pulse code modulation
DMO	dip move-out
DWHT	discrete Walsh-Hadamard transform
DWPT	discrete wavelet packet transform
DWT	discrete wavelet transform
DWTC	discrete wavelet transform coding
EBCDIC	Extended Binary Coded Decimal Interchange Code
EC	entropy coding

EZC	embedded zero-tree coding
FB	filter bank
FIR	finite impulse response
GenLOT	generalized lapped orthogonal transform
HC	Huffman coding
IBM	Industrial Business Machines, Corp.
IEEE	Institute of Electrical and Electronics Engineers
i.i.d.	independent and identically distributed
IIR	infinite impulse response
JPEG	Joint Photographic Expert Group
LOT	lapped orthogonal transform
mse	mean-square error
NMO	normal move-out
pdf	probability density function
PGS	Petroleum Geo-Services
PR	perfect reconstruction
QMF	quadrature mirror filter
RAI	relative acoustic impedance
RIH	reel identification header
RLC	run-length coding
rms	root-mean-square
rmse	root-mean-square error
SBC	subband coding
SCDI	Seismic Compression Diagnostic Initiative
SEG	Society of Exploration Geophysicists
SGI	Silicon Graphics, Inc.
SNR	signal-to-noise ratio
SQ	scalar quantization
SQNR	signal-to-quantization-noise ratio
SUN	Sun Microsystems, Inc.
SVD	singular value decomposition
TAR	true amplitude recovery
TC	transform coding
TIH	trace identification header
TWT	two-way traveltime
VQ	vector quantization
WEMR	wave equation multiple rejection

Chapter 1

Introduction

Interpreters of seismic data struggle with the limitations of their hardware and software every day. The archiving and transporting of seismic data volumes require huge transmission capacities and large storage mediums. A large seismic survey typically comes to more than one terabyte; hence the seismic industry must search for effective ways of handling data. To work around these problems, geoscientists have traditionally decreased the size of the seismic data volume for their interpretation. Examples include reduction of the temporal range and/or resampling in time, reduction of the spatial range and/or resampling in space, and clipping of the amplitudes, thereby reducing the dynamic range.

Different from conventional or conservative approaches, compression is in this thesis considered as the most desirable way of improving the efficiency of transmission and storage of seismic data. Given a certain limited resource of storage capacity or transmission bandwidth, lossy seismic data compression is without question a much better alternative than non-sophisticated data reduction methods. Contrary to traditional data reduction methods, seismic data compression preserves the dynamic range and can increase the resolution of the seismic signal. Depending upon the required fidelity of the reconstructed (decompressed) seismic data, the goal is to obtain compression ratios of 10:1-20:1 for prestack seismic data in the case of 2-D compression algorithms, or even 50:1-100:1 in the case of 3-D compression algorithms. The ability to compress seismic data for example 10:1 means that 10 times more data can be stored in the space that it takes to store the original data set.

Data compression makes the storage of seismic data more efficient, and has a potential of reducing the time for network and satellite transmission of seismic data from hours to minutes. Hence, a significant reduction in cost is achieved. For example, a compression ratio of 50:1 reduces one terabyte of seismic data to 20 gigabytes. Taking the cost of storage per megabyte of seismic data on robotic tape to be $0.20 per year, the storage price would be reduced by 196,000 dollars per year per terabyte of raw data. Furthermore, assuming a satellite transmission cost equal to $2.0 per megabyte of seismic data, the transmission cost of the same compression example would be decreased from 2.0 million dollars to 40,000 dollars per terabyte of raw data.

Two main types of compression techniques exist; lossless and lossy compression. No errors are introduced into the data by *lossless compression* and the seismic data set can be reconstructed without any loss of information. The least significant bits (containing mostly background noise) and the most significant bits in this case are given equal weight of importance. Since all bits are reconstructed exactly, lossless compression of seismic data can only offer compression ratios in the order of 2:1. Compression of text files (e.g., trace identification headers), on the other hand, can achieve compression ratios in the order of several hundreds to one by lossless methods (Røsten et al., 1999c). A text file contains frequently repeated letters and patterns of letters while a typical seismic data file is far less structured (Reiter and Heller, 1994). During *lossy compression* of seismic data some errors are introduced. Due to the introduction of compression noise, lossy methods generally offer higher compression ratios than lossless methods. The characteristics of the compression noise depend on the compression ratio. A high compression ratio (much higher than 10:1) gives a high level of compression noise. At the same time, the compression noise becomes more and more coherent (i.e., non-white) as the compression ratio increases.

Throughout all parts of the seismic industry, from acquisition to processing and interpretation, users are careful not to employ procedures that introduce noise into the seismic data set. In general, compression of prestack seismic data at the acquisition phase can be dangerous since all processing procedures are performed after compression and decompression; compression at later stages of processing is safer. Nevertheless, as such, lossy seismic data compression can be seen as just one of the noise sources in the acquisition and processing of seismic data. What "simply" remains before seismic data compression becomes widely used is that the effects of compression noise on seismic data must be thoroughly understood and documented. With an understanding of the associated drawbacks it is expected that the need for reduced turnaround times, from acquisition to processing and interpretation, will push the use of seismic data compression as a routine component of standard survey operations.

1.1 Background and objective

From a very simplistic point of view, seismic signals can be viewed as a combination of three types of components: pure geophysical or geological information, redundancy in the signal representation, and noise or alterations arising from different kinds of sources. This combination results in the statement

$$\textit{seismic signal} = \textit{information} + \textit{redundancy} + \textit{noise}.$$

Entropy is a measure of the amount of information in a signal. Redundancy is a repetition of information such as the same measurements made several times (perhaps in several ways), but also for example multiple reflections in the case of plane, homogeneous layers. Multiple reflections can, dependent on the application, also be regarded as information or (source-generated coherent) noise. Each and every task capable of separating information, decreasing redundancy, and suppressing noise is generally not performed by one single method, but commonly involves many seismic data processing tools. Besides, many seismic applications including analysis, denoising, processing, and data management (e.g., data access, visualization, and compression) commonly provide a focusing of the information. By reducing the dynamic range of the seismic data in some way or another, it is possible to discard redundancy and noise in the seismic data set. Discrete wavelets provide such a representation. They have been applied to seismic data analysis (Foster et al., 1997), to denoising of seismic data (Miao and Cheadle, 1998), to migration of seismic data (Dessing and Wapenaar, 1994, 1995; Wu and McMechan, 1998), and to seismic data compression (Bosman and Reiter, 1993; Donoho et al., 1995; Chen, 1995).

Discrete wavelets are based on the usage of a lowpass and a highpass filter, employed in an iterative scheme leading for example to the discrete wavelet transform (DWT) or to the discrete wavelet packet transform (DWPT). A decomposition scheme involving several stages of filters and sampling operators (i.e., up-sampling and down-sampling) is generally called a *filter bank* system, and DWT and DWPT are sub-class of filter banks (Ramstad et al., 1995). Filter banks are more universal than (orthogonal or biorthogonal) discrete wavelets by involving more than 2 filters, having several different iterative schemes, and being non-unitary (i.e., neither orthogonal nor biorthogonal).[1.1] Since discrete wavelet transforms represent a special case of filter banks, they may be inferior to non-unitary filter banks as shown for seismic data compression by Røsten et al. (1996, 1997a,b). Non-unitary filter banks have also been

[1.1] Biorthogonal filter banks are strictly speaking also non-unitary.

applied to denoising (i.e., white-noise attenuation) of seismic data by Duval and Røsten (2000) and to migration of seismic data by Røsten and Ramstad (1998) and Røsten et al. (1998a,b).

The main topic addressed in this thesis is lossy compression of prestack seismic data using *subband coding* (SBC; Ramstad et al., 1995), the main focus being optimization of the method with an investigation of the effects of compression noise on seismic processing and interpretation.

The principles of SBC are: At the encoder side, the input signal is decomposed into M spectral subbands by an analysis filter bank to reduce the redundancy. The analysis filter bank decorrelates the input signal, and concentrates the energy of the input signal into a minimum number of subband channels. After subband decomposition, the subband samples in branch or channel number m for $m = 0, 1, \ldots, M-1$ are critically down-sampled by α_m (every α_mth subband sample is retained) with $\alpha_m \in \mathbb{Z} \geq 2$ and

$$\sum_{m=0}^{M-1} 1/\alpha_m = 1, \tag{1.1}$$

to keep the total number of subband samples unchanged compared to the original signal representation. Then the down-sampled subband samples are quantized by scalar quantization to allow for high compression ratios. The scalar quantization stage gives an approximation of the subband samples, and solely introduces the compression noise.[1.2] Finally, the quantized subband samples are entropy coded (thus compression is achieved). At the decoder side, the M spectral subbands are entropy decoded and dequantized by inverse scalar quantization. Then the subband samples in branch or channel number m are up-sampled by α_m (that is inserting $\alpha_m - 1$ zeros between every dequantized subband sample), and the signal is at the end reconstructed by a synthesis filter bank. In this dissertation only *uniform* filter banks, where all subband channels have equal bandwidth, i.e., setting all α_m's to M, will be considered.

[1.2] The analysis and synthesis filter banks are assumed to be a perfect reconstruction system.

1.2 Fundamentals

In this section, the basics of seismic data acquisition and processing are presented. The intention is not to give a comprehensive description of these issues, but rather to present the most important parts of seismic data acquisition and processing required to understand the terminology used in this dissertation. For details, the interested readers are referred to Yilmaz (1987) and Sheriff and Geldart (1995).

Exploration seismology includes the fields of data acquisition, processing, and interpretation. The main objective of a reflection survey is to map the depth, dip, and strike of interfaces which usually are parallel to the bedding, and lateral or horizontal changes in reflections. A second objective is to define stratigraphic variations from normal move-out measurements or from the amplitude and waveshape of reflection events.

1.2.1 Seismic data acquisition and preprocessing of data

Offshore, seismic data are collected by seismic vessels. Air guns filled with highly compressed air are commonly used as an energy source generating P-waves or acoustic waves. A P-wave is a body wave in which particle motion is in the direction of the propagation, and is the type of seismic wave recorded in conventional marine seismic exploration. A P-wave is also called a longitudinal wave or pressure wave. The air-gun array is towed behind the survey vessel on a suitable frame. When the source is fired, the pressure wave propagates down through the water layer and into the earth below the sea floor. When a wave hits an interface separating two different geologic layers, some of the wave is reflected and the rest of the wave is transmitted. The reflected wave travels upwards and is finally measured by pressure-sensitive hydrophones deployed in the streamer or cable towed behind the vessel. Each hydrophone records pressure-wave amplitude reflections as a function of two-way traveltime (TWT).

Seismic data are recorded and digitized in rows of samples; samples at the same time at consecutive hydrophone or receiver groups. Demultiplexing is the first *preprocessing* task which involves sorting the data into columns of samples; all the time samples in one receiver group followed by all the time samples of the next receiver groups. The record for each shot from each receiver group is now called a trace. Preprocessing also involves trace editing; noisy traces, for example traces with transient glitches, are zeroed. Finally, field geometry and acquisition parameters are incorporated with the seismic data. Based on surveying information, coordinates of shot and receiver-group locations for all traces are stored on trace identification headers. Geometry assignment is one of the most important aspects of seismic data processing, and many types of seismic data processing problems arise from incorrectly setting up the field geometry.

Specifically, the seismic data set considered in this dissertation is a 2-D surface line acquired offshore central Norway during 1992. The source was a conventional air-gun array with a total volume of 1580 cubic inches ($\approx$ 25900 cm^3). The shot interval was 18.75 m, the source depth was 5 m, and the total number of shots in the data set was 1101. Furthermore, the streamer had 240 receiver groups with a group distance of 12.5 m, and the streamer depth was 8 m. The source-receiver-offset is defined as the distance from the center of the source array to the center of the receiver group. The distance between the center of the source array and the center of the *first* group (near offset) was 108.5 m. Finally, the record length (i.e., TWT) was 4000 ms with sample interval 2 ms. Hence, the number of samples per trace in the data set is 2001.

As further preprocessing, the data set was conveniently decimated by a factor of 2 in time applying anti-alias filtering, increasing the time sample interval from 2 to 4 ms and reducing the number of samples per trace from 2001 to 1001 (see Figure 1.1).

1.2.2 Domains or organizations of seismic data

The simplest 2-D shooting configuration will suffice to illustrate the four domains commonly used to process and analyze seismic data. Figure 1.2 illustrates a standard off-end recording geometry assuming a single plane reflector. Off-end spread configuration means that all the receiver groups are on one side of the shot, for example to the right of the shot. For each shot or recording in 2-D profiling, the coordinates of the shot point and each receiver group along the lateral distance are surveyed.

Figure 1.3 is known as a *stacking chart* and illustrates a useful technique for visualizing the interrelationships along overlapping recordings in a conventional 2-D seismic survey. By plotting shot coordinate s against receiver coordinate g in a 2-D orthogonal coordinate frame, the overlapping recording geometries can be visualized. The stacking chart can help us understand the different domains or organizations of the data that are used in seismic data handling. For example, a horizontal slice parallel to the g axis at a fixed s coordinate represents a common shot gather (CSG) or the shot domain, this is the way the data are collected [see also Figure 1.1 and Figure 1.4 (a)]. A vertical slice at a

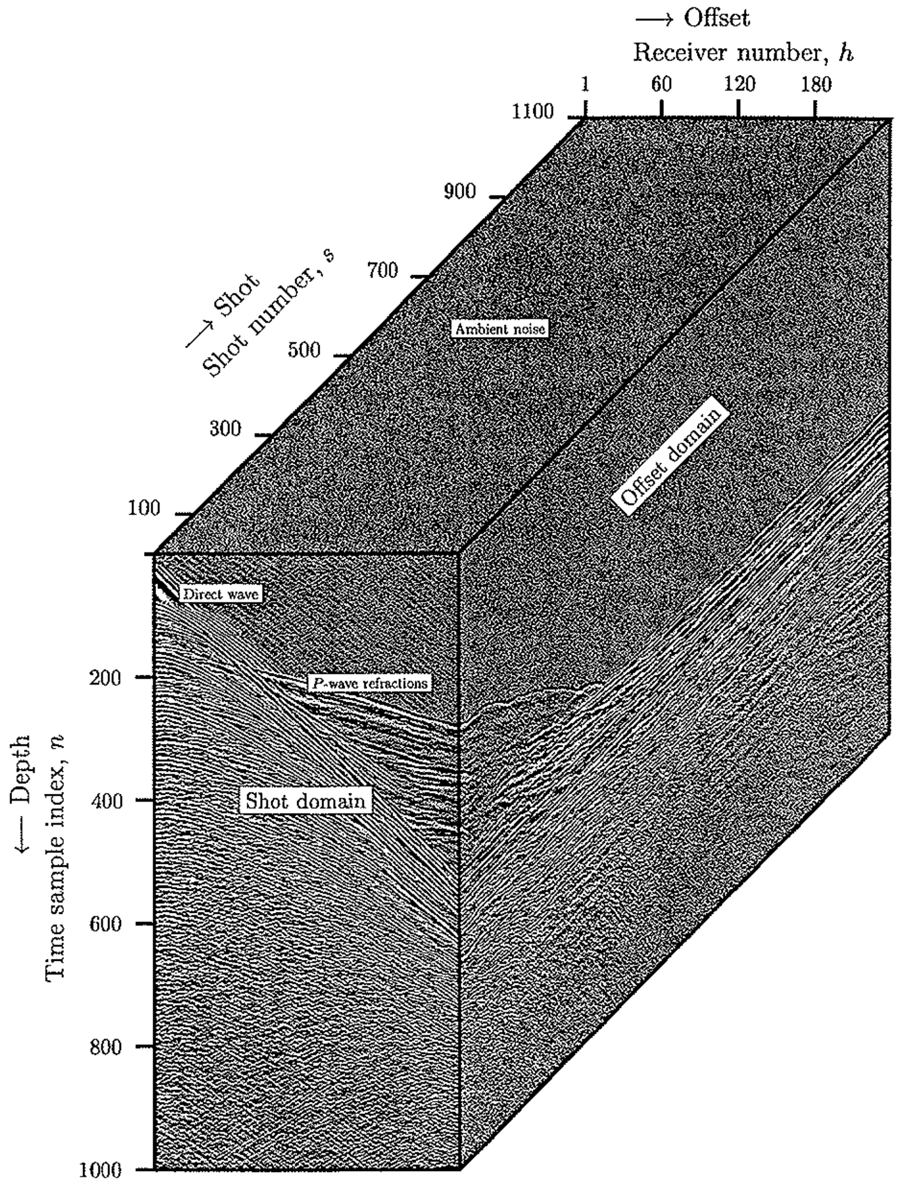


Figure 1.1: Cube of the data set consisting of 1101 shots, 240 receivers per shot, and 1001 samples per trace. The first time sample index and the first shot number start at 1.

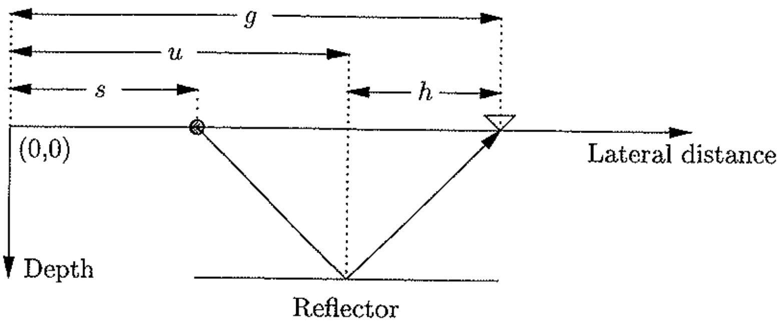


Figure 1.2: Seismic data acquisition is done in shot-receiver (s, g) coordinates, and mid-point-offset (u, h) coordinates is an alternative coordinate representation. The shooting direction is to the left and the ray path is given by the arrow. The gray circle denotes the shot and the white triangle denotes a receiver group.

fixed g coordinate represents a common receiver gather (CRG) or the receiver domain [see Figure 1.4 (b)], which is a seismic experiment in which many shots at different coordinates s are recorded by a single receiver group at location g. There are two other domains that are very important as well. They are represented by the rotated axis corresponding to the coordinate transformation into the common offset axis

$$h = (g - s)$$

and the common mid-point axis

$$u = (s + g)/2.$$

The offset axis h measures (half) the source-to-receiver distance, and the mid-point axis u measures the average source and receiver distance along the seismic profile. A collection of traces parallel to the mid-point axis at at fixed h is called a common offset gather (COG) or the offset domain [see also Figure 1.1 and Figure 1.4 (c)]. Finally, a collection paralleling the offset axis at a fixed u is called a common mid-point (CMP) gather [see Figure 1.4 (d)], the latter representing the CMP domain.

All four domains are useful representations for different aspects of seismic signal processing and analysis. Nevertheless, the CMP domain is the primary one for conventional seismic data processing. An important observation about the different domains of seismic data is that the CSG configuration of Figure 1.4 (a) represents the only physical experiment actually performed in the seismic

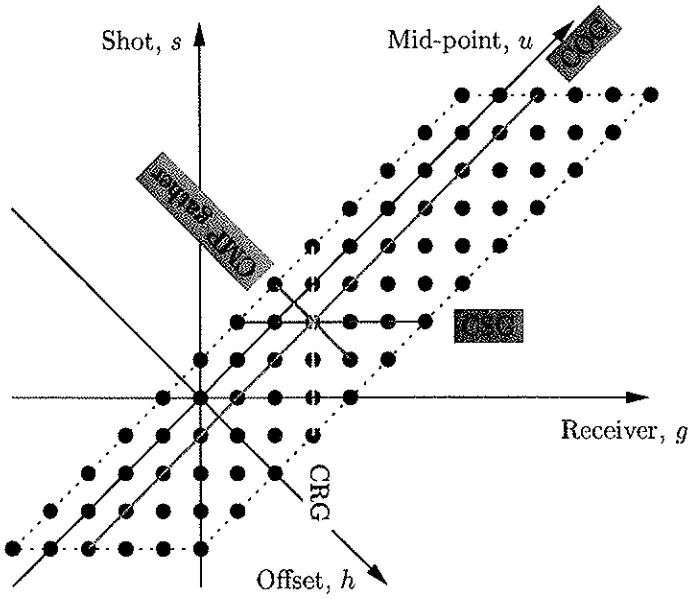


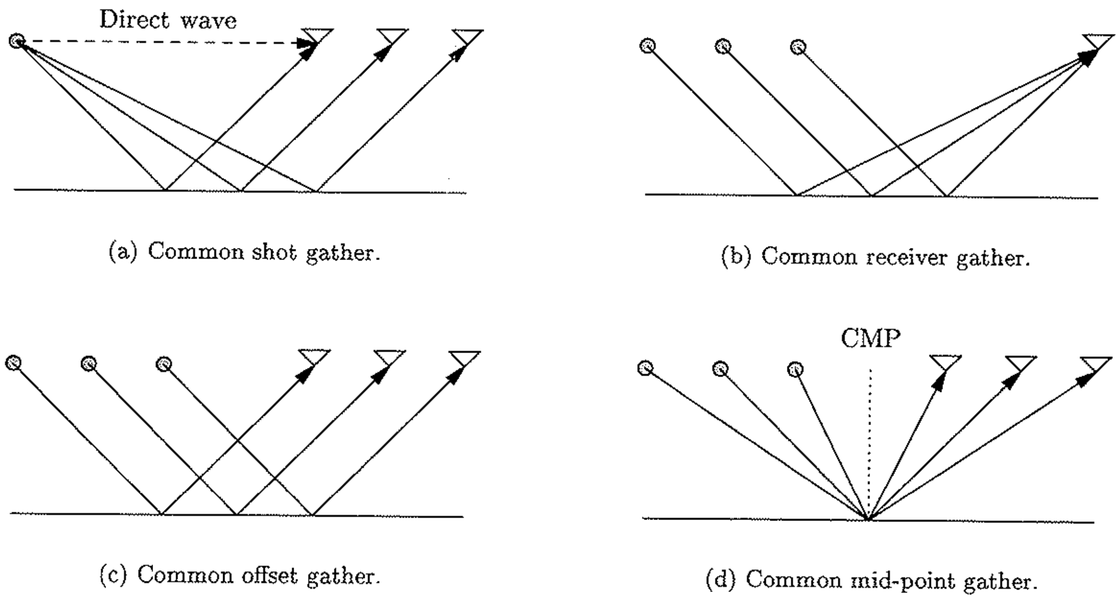
Figure 1.3: Stacking chart. Each black dot represents a single trace with the time axis perpendicular to the plane of the page. Shot-receiver (s, g) and mid-point-offset (u, h) coordinates are superimposed with the (u, h) plane rotated 45 degrees in the clockwise direction with respect to the (s, g) plane.

data acquisition phase. The other three domains are simply reorganizations of data performed after acquisition by sorting traces according to the domains.

As the distance between source and receiver increases, the travel path for reflection lengthens and the arrival time for a reflection event at the detector increases. The *move-out* is the difference in arrival time at different receiver positions. For a single horizontal reflector separating two homogeneous half-spaces, the simplest one of all earth models, the TWT-distance relationships or move-out of the reflector become horizontally aligned for a COG and hyperbolically shaped for the three other types of gathers. Thus, in this case, the offset domain is the most attractive one for seismic data compression due to higher redundancy compared to the shot domain (see Figure 1.1).

1.2.3 Representation of frequency and wavenumber

The analysis and processing of seismic data can be performed in several other domains different from those previously mentioned. The transformation of seismic data from time and space to frequency and wavenumber (i.e., spatial frequency), respectively, is sometimes appropriate for several reasons (e.g., the



(a) Common shot gather.

(b) Common receiver gather.

(c) Common offset gather.

(d) Common mid-point gather.

Figure 1.4: Useful domains of seismic data. The common shot domain (a) represents the only physical experiment actually performed in the seismic data acquisition phase. The common receiver domain (b) is not important for this thesis and is only shown for completeness. The common offset domain (c), on the other hand, is of highly importance for this thesis since seismic data compression is applied on common offset gathers. Finally, the CMP domain (d) is the primary one for seismic data processing.

issue of implementation and the consideration of artifacts). The transformation is usually done by the discrete (or fast) Fourier transform applied separately in time and space. The frequency-wavenumber domain is simply the result of a 2-D discrete Fourier transform of a seismic record or a seismic section. Applications include for instance frequency-wavenumber dip filtering, anti-aliasing filtering, and migration. The two other combinations of frequency-space domain and time-wavenumber domain can also be suitable for some applications.

1.2.4 Classification of noise

The term signal is used for any event on a seismic record which is used to obtain information about the subsurface. Noise, on the other hand, is an unwanted signal. Noise can be classified into two parts:

- Coherent or colored noise;
- Incoherent or white noise.

Coherent noise is correlated from trace to trace on a record, while incoherent noise is not.

Alternatively, noise can be classified into *source-generated noise* or *ambient noise*. Examples of source-generated coherent noise components in marine seismic work are multiples. Multiples are reflected waves that have undergone more than one reflection, and multiples are usually considered as an important source of noise. Unless the water or sea surface is quite rough, it is an almost perfect reflector for upcoming seismic waves. In many areas, the water bottom is also a hard reflector, resulting in a large fraction of the source energy being trapped and reverberating in the water layer, generating water-bottom multiples. In addition to multiples trapped entirely in the water layer, any reflected wave can have one or more water-layer reverberations added to it at the source or receiver end of its path, creating so-called peg-leg or inter-bed multiples. Ambient noise is the same as background noise. Examples of ambient coherent and incoherent noise components in marine seismic work are vessel noise and wave noise, respectively.

The most troublesome noise is source-generated coherent noise. The signal-to-noise ratio (SNR) for ambient noise, whether coherent or incoherent, can be increased by increasing the source energy or by enlarging the fold (to be explained later). Source energy has no effect on SNR for source-generated coherent noise and this type of noise is most effectively attenuated by for example seismic data processing techniques, e.g., methods for attenuation of multiple reflections.

From now on, ambient noise is simply treated as incoherent noise such as wave noise (see Figure 1.1), and source-generated noise is simply regarded as coherent noise. Noise is therefore classified into either (i) source-generated coherent noise (or compression-produced coherent noise at high compression ratios) or (ii) ambient incoherent noise (or compression-generated incoherent noise at low compression ratios).

1.2.5 Categorization of seismic signal

The recorded seismic signal in marine exploration, ignoring redundancy and noise, consists of different types of signal contributing to the information about the subsurface. Some types of signal are more important than other. The primary reflections are the main signal component, and the other signal types are sometimes classified as noise since they usually are unwanted. For this thesis a definition of the four most significant classes of signal are needed (Sheriff, 1991):

1. direct wave;
2. (primary) reflections;
3. refractions;
4. diffractions.

The *direct wave* [see Figures 1.1 or 1.4 (a)] corresponds simply to the source energy which travels directly from the source to the receivers. The direct wave is easily attenuated by a mute, at least for small offsets. A *reflection* is the energy or wave from a seismic source which has been reflected according to Snell's law from a contrast in elastic parameters (i.e., density, P-wave velocity, and shear-wave or S-wave velocity) or series of contrasts within the earth. The objective of most reflection seismic work is to determine the location and attitude of reflectors from measurements of the traveltime of primary reflections and to infer the geologic structure and stratigraphy. *Refractions* are also called head waves. A head wave is a wave characterized by entering and leaving a high-velocity medium at the critical angle defined by Snell's law. P-wave refractions are effectively muted, at least for large offsets (see Figure 1.1). Finally, *diffractions* are the bending of wave energy around obstacles without obeying Snell's law. Such events result at the termination of reflectors (for example at faults) and are characterized on seismic records and sections by a distinctive curved alignment. When correctly migrated, a simple diffraction collapses at the location of the diffracting point.

1.2.6 Seismic data processing

The preprocessed seismic data must be processed further to give useful information. First static corrections are applied to the seismic data to compensate for the effects of variations in elevation, weathering thickness, weathering velocity, and reference from a floating datum to a final datum. The remaining or primary processing stages in reflection seismic methods in their usual order of application are:

- true amplitude recovery;
- predictive deconvolution;
- attenuation or rejection of multiple reflections;
- migration;
- stacking.

The seismic data processing techniques may operate in any one of the domains presented in Figure 1.4. For the data set used in this thesis, geometrical spreading and absorption are compensated for by applying true amplitude recovery in the shot domain. Predictive deconvolution is applied in the shot domain as well. Ideally, the process of predictive deconvolution improves the *temporal resolution* of seismic data by compacting the seismic wavelet and attenuating ringing energy. The seismic wavelet is equal to the source signature convolved with the source and receiver ghosts and convolved with the instrument response. In order to give an acceptable result, traditional deconvolution, based on a causal infinite impulse response (IIR) or inverse filter, requires the seismic wavelet being minimum-phase. A more advanced deconvolution technique can for example be found by convolving a maximum-phase finite impulse response (FIR) filter with a minimum-phase IIR filter, i.e., so-called mixed-phase deconvolution (Porsani and Ursin, 1998; Ursin and Porsani, 2000). Furthermore, deconvolution assumes seismic data without, or at least small amounts of, noise; noise components can have harmful effect on predictive deconvolution (Duarte, 1992). In this thesis, two different multiple attenuation techniques are applied. Attenuation of water-bottom multiples (actually performed before true amplitude recovery) is done in the shot domain using the so-called wave-equation-multiple-rejection (WEMR; Wiggins, 1988) method. Rejection of peg-leg or inter-bed multiples using the parabolic Radon transform (Hampson, 1986), on the other hand, is performed after predictive deconvolution in the CMP domain with normal move-out (NMO; explained in more detail below) correction

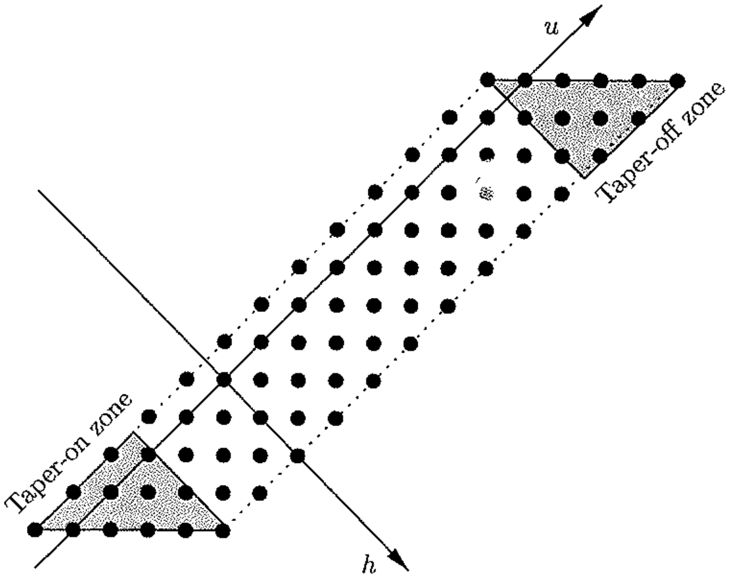
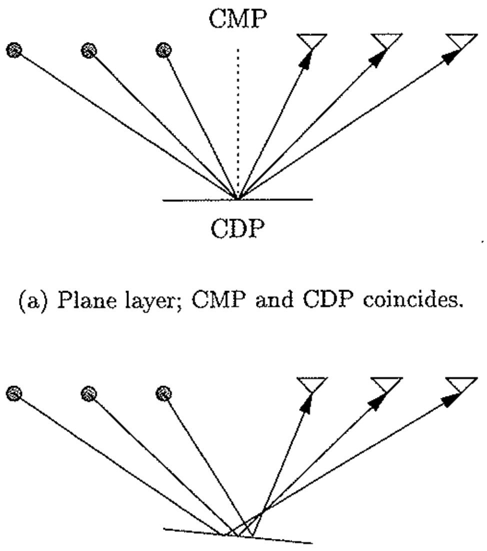


Figure 1.5: Taper-on and taper-off zones in the CMP domain.

applied to the CMP gathers. Then, after inverse NMO correction, prestack migration is done in the offset domain. Migration moves dipping reflectors into their true subsurface positions and collapses diffractions, thereby improving the *spatial resolution*. At the end, stacking is applied in the CMP domain.

If the shot-receiver spread in a multi-channel reflection survey is moved forward in such a way that any two reflected ray paths do not sample the same point on a subsurface reflector, the survey coverage is said to be single-fold. A conventional CMP recording technique, on the other hand, uses redundant recording to improve the SNR. In CMP profiling, which has become the standard method of 2-D multi-channel seismic surveying, a set of traces recorded at different offsets contains reflections from a CMP on the reflector. The advantages of CMP surveying are many. First, the CMP gather represents the best possible data set for computing the velocity field from the NMO effect (discussed in the subsequent section). Second, with accurate velocity information the move-out can be removed from each trace of a CMP gather to produce a set of traces that may be summed algebraically. The traces within a CMP gather are summed to yield a single stacked trace, and the collection of several side by side stacked traces gives a seismic stack image. The data before and after stacking are called prestack and poststack seismic data, respectively. The CMP stack or simply the stack is generally acknowledged as the most im-



(a) Plane layer; CMP and CDP coincides.

(b) Dipping layer; CMP and CDP do not converge.

Figure 1.6: CDP reflection profiling for plane and dipping layers.

portant data-processing application in improving data quality, i.e., increasing SNR. CMP stacking effectively attenuates ambient incoherent noise, but also source-generated coherent noise such as multiples. This is because reflected signals and coherent noise usually have different (stacking) velocities.

The fold of the stacking refers to the number of traces in the CMP gather. The higher the fold is, the higher the gain in SNR is achieved. At the beginning and ending of a CMP gather, the fold of coverage is gradually built up and dropped off, respectively. These two zones are known as taper on and taper off (see Figure 1.5).[1.3] The CMP gathers having maximum fold of coverage are said to be full-fold. The fold for the data set used in this thesis is given by the number of receiver groups multiplied by the receiver-group distance divided by two times the shot interval, that is $F = (240 \times 12.5)/(2 \times 18.75) = 80$.

A CMP coincides with the common depth-point (CDP) in the case of plane, homogeneous layers [see Figure 1.6 (a)]. Strictly, the CMP principle breaks down in the presence of dip because the CDP no longer converges with the CMP and the reflection point differs for rays traveling to different offsets [see Figure 1.6 (b)]. Dip move-out (DMO) correction is a process that creates

[1.3] A CRG has taper-on and taper-off zones as well, but not CSGs and COGs.

apparent CDP gathers with the feature that the NMO for reflectors from a dipping bed no longer depends on the dip angle. Note that the term CDP frequently, but uncritically, is used instead of CMP. Both NMO and DMO are key elements of modern seismic data processing, and a non-mathematical discussion of the nature and action of these procedures is presented by Liner (1999). NMO, DMO, inverse NMO, CMP stacking, and poststack migration are conventionally used instead of prestack migration if the structure and velocity variation are not too ill-natured. However, this efficient sequence of operations breaks down and fails to give a good image in the case of structural complexity and/or lateral velocity variation, and a single grand process (prestack migration) is required to achieve a geologically meaningful image.

1.2.7 Velocity analysis

The estimate of propagation velocity in the reflection seismic method is essential to the effective imaging of the subsurface and in particular for NMO and migration. The velocity is a property of the medium and depends on the type of waves, for example the P-waves in a conventional reflection seismic survey. A common simplification is the assumption of isotropic medium which means that the velocity does not depend on the direction the wave is traveling. Dispersion is a variation of velocity with frequency, but the dispersion of seismic body waves is very small under most circumstances.

Seismic data provide an indirect measurement of velocity. A large number of different types of velocities can be derived, such as interval, average, root-mean-square (rms), NMO, stacking, and migration velocities. However, the velocity that most commonly is derived from seismic data is the velocity that yields the best stack, i.e., the stacking velocity. It is determined by *velocity analysis* and is the value used for CMP stacking. Assuming a layered media, stacking velocity is related to NMO velocity. This, in turn, is related to rms velocity, from which the average and interval velocities are derived.

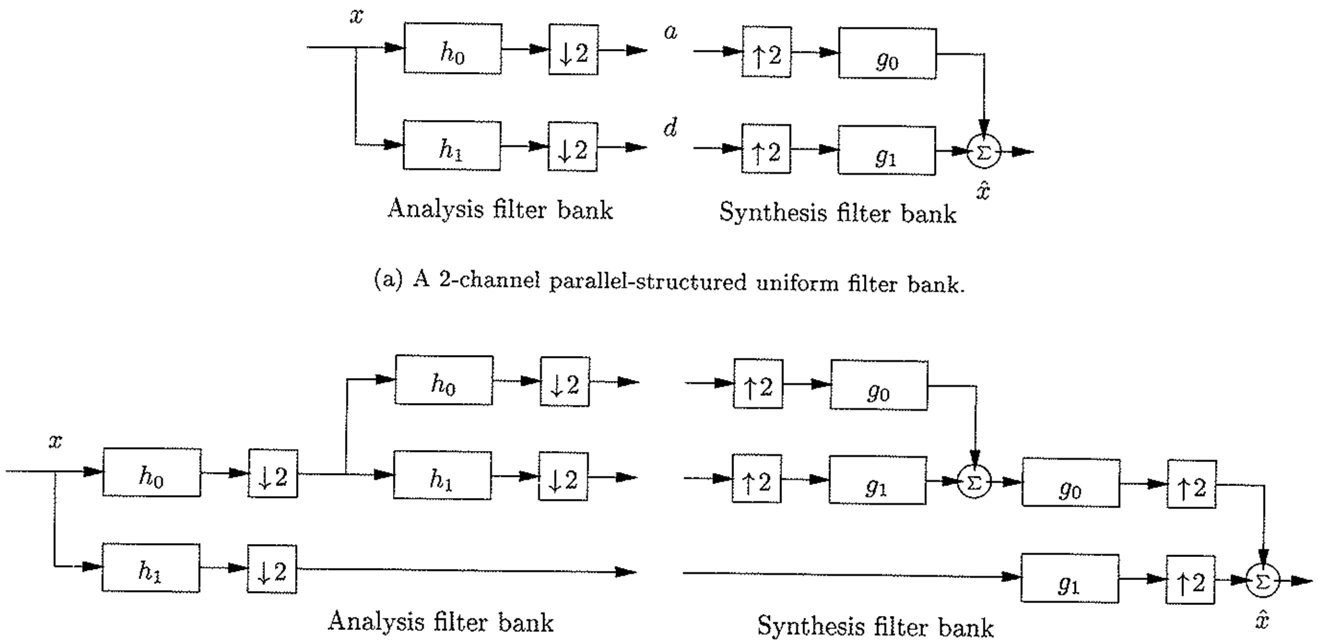
The specific details of velocity estimation are not to be discussed. Most of the traditional velocity estimation methods are based on manual inspection of semblance plots of the seismic data, that is so-called velocity picking. As previously mentioned, the NMO for CMP gathers is the best basis for determining velocities from seismic data. The velocity required to correct for NMO is the NMO velocity. For a single horizontal layer, the NMO velocity is equal to the velocity of the medium above the reflector. For a dipping reflector, the NMO velocity is equal to the medium velocity divided by the cosine of the dip angle. The NMO velocity is the velocity for NMO correction in the limit as the source-receiver-distance becomes small. In the limit as the offset approaches zero, the stacking velocity approaches the NMO velocity.

An important observation is that the velocities required by stacking and migration are not necessarily the same. Similar to the NMO velocity, for data collected parallel to the dip direction of a single dipping reflector, the stacking velocity is the velocity of the medium above the reflector divided by the cosine of the dip angle. On the contrary, the migration velocity is the velocity of the medium itself. In other words, stacking velocity is sensitive to dip angle, while migration velocity is not. A widespread, conventional approach to determine dip-independent stacking velocities is to apply velocity analysis on CMP gathers after NMO (using initial NMO or initial stacking velocities), DMO, inverse NMO, and prestack migration of all common offset gathers. This method is employed in this dissertation, and the estimated stacking velocity is used for NMO while a *smoothed* version of the estimated stacking velocity is used for prestack migration to avoid artifacts.

1.3 Filter banks

Figure 1.7 (a) shows a 2-channel analysis-synthesis filter bank system. At the analysis side, the input signal x is filtered by a lowpass filter h_0 and a highpass filter h_1. The outputs of the filters are critically down-sampled by 2, i.e., every second sample is removed. The analysis filter bank splits the input signal into a, a coarse and smoother approximation (lowpass filtered version) of x, and into d, a detail signal (highpass filtered version of x). The resulting decomposition is non-expansive: the total number of samples after the analysis filtering is the same as the number of input samples due to the down-sampling procedure. At the synthesis side, a and d (called subband samples) are up-sampled by 2 (one zero is inserted between every subband sample). Then the two up-sampled signals are filtered by lowpass and highpass synthesis filters, denoted g_0 and g_1, respectively. The reconstructed signal $\hat{x}$ finally follows by a summation at the end. The signal x can be perfectly reconstructed, i.e., $x = \hat{x}$ if the analysis and synthesis filters obey a so-called perfect reconstruction (PR) condition. That is, the synthesis filter bank cancels aliasing, amplitude, and phase distortions introduced by the analysis filter bank.

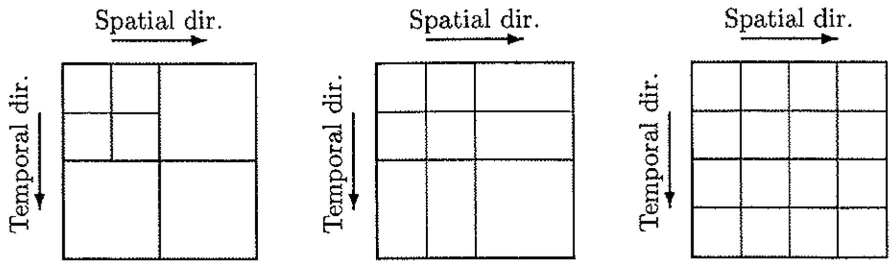
The special case of DWT is based on a repeated tree-structured lowpass analysis filtering of the approximation a, and results in a non-uniform filter bank with a so-called dyadic or octave-band decomposition. Figure 1.7 (b) shows a 2-level tree-structured dyadic filter bank system. In the 2-D case,



(a) A 2-channel parallel-structured uniform filter bank.

(b) A 2-level tree-structured dyadic filter bank.

Figure 1.7: 2-channel parallel-structured and 2-level tree-structured maximally decimated filter banks. The symbols $\downarrow 2$ and $\uparrow 2$ denote down-sampling and up-sampling by 2, respectively.



(a) Dyadic; type-1. (b) Dyadic; type-2. (c) Uniform.

Figure 1.8: 2-level dyadic type-1 decomposition, 2-level dyadic type-2 decomposition, and 4-channel uniform decomposition.

one can apply a 1-level DWT decomposition first in the temporal direction and second in the spatial direction, and then iterate the 1-level DWT analysis filtering procedure $S-1$ number of times on the lowpass-lowpass subband [type-1; see Figure 1.8 (a)]. An alternative is to apply an S-level DWT decomposition first in time and second in space [type-2; see Figure 1.8 (b)]. Of the two different types of DWT schemes, type-1 is more known than type-2.

A general structure of an $S+1$-channel parallel-structured filter bank system is given in Figure 1.9. The 1-D input signal x is split into $S+1$ number of subbands by an analysis filter bank. The sampling coefficients α_m for $m = 0, 1, \dots, S$ and $\alpha_m \in \mathbb{Z} \geq 2$ are subject to the condition

$$\sum_{m=0}^{S} 1/\alpha_m = 1 \tag{1.2}$$

in order to keep the decomposition non-expansive. The special case $S = 1$, i.e., a 2-channel filter bank with $\alpha_0 = \alpha_1 = 2$, represents a 1-level DWT. An $S+1$-channel filter bank with

$$\alpha_0 = 2^S \tag{1.3a}$$

and

$$\alpha_m = 2^{S+1-m}, \quad 1 \leq m \leq S, \tag{1.3b}$$

has equivalent decomposition appearance as an S-level DWT.

In the case of 2-channel (or $S+1$-channel for $S+1 > 2$) filter banks, the filters can be divided into three classes:

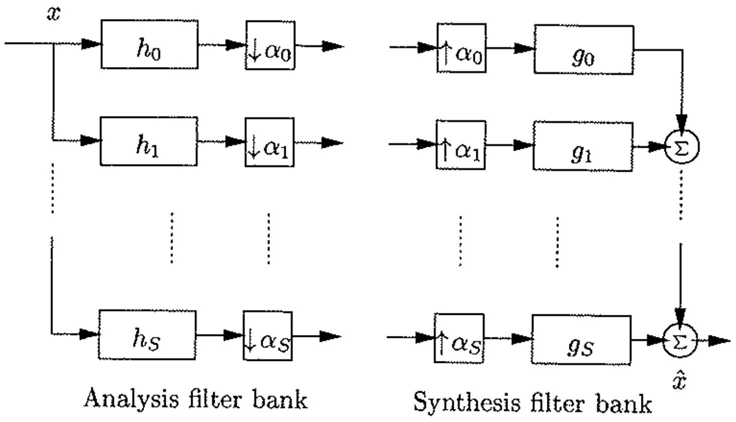


Figure 1.9: An $S+1$-channel parallel-structured filter bank.

- orthogonal PR filter banks: they generate an orthogonal basis for signal decomposition.
  - h_1, g_0, and g_1 are derived from h_0;
- biorthogonal PR filter banks: they generate two bases where the h_m's are orthogonal to the g_m's, but the two bases are not orthogonal themselves.
  - g_1 and h_1 are derived from h_0 and g_0, respectively;
- other non-unitary filter banks or filter banks without restrictions.[1.4]

The choice of the h_m and g_m filters for $m = 0, 1$ is crucial. Please refer to the book by Strang and Nguyen (1996) for a comprehensive survey of the construction of orthogonal and biorthogonal 2-channel PR filter banks. Orthogonal bases organized in non-overlapping blocks, like the discrete Karhune-Loève transform (DKLT), generally provide better separation between signal and (white) noise components (Therrien, 1992). For instance, Ioup and Ioup (1998) used an orthogonal Daubechies wavelet for denoising of seismic data, and showed its advantages over standard discrete Fourier transform filtering. Nevertheless, the orthogonality constraint is a very strong condition in 2-channel filter bank design. For instance, it is not possible to have both orthogonal PR and linear-phase filters except for trivial cases like for example the orthogonal Haar wavelet. The orthogonality constraint is relaxed for biorthogonal and other non-unitary filter banks. As a consequence, in the 2-channel case, biorthogonal filter banks may simultaneously be PR and have

[1.4] Non-unitary filter banks need not to be PR.

linear-phase – non-trivial – filter coefficients. The linear-phase property is commonly used to avoid for example artifacts such as edge effects in data compression. Miao and Cheadle (1998) used the popular "9,7" biorthogonal wavelet (i.e., 9 and 7 number of filter taps in h_0/g_1 and g_0/h_1, respectively) in noise attenuation of seismic data. At the same time, many data compression applications involve the same "9,7" biorthogonal wavelet (e.g., Donoho et al., 1995; Vassiliou and Wickerhauser, 1997) or non-unitary linear-phase near-PR filter banks (e.g., Røsten et al., 1996, 1997a).

Some authors (see e.g., Vermeer et al., 1996), have pointed out that the DWT might not be the best suited decomposition scheme since it lacks accuracy in the high frequency subbands. Furthermore, a 2-channel filter bank intuitively possesses less degrees of freedom than filter banks having three or more channels. Simultaneous, orthogonality and linear-phase constraints in the $S+1$-channel case for $S+1>2$ are easier to meet since more filters add more coefficients to be tuned. By relaxing the orthogonality or biorthogonality constraint even greater degrees of freedom will arise. In the rest of the thesis, the main focus is on M-channel parallel-structured uniform decompositions using non-unitary linear-phase near-PR filter banks, setting all α_m's to $M=S+1$ (see Figure 1.10). Specifically, the number of channels used is $M=8$. Figure 1.8 (c) illustrates a uniform decomposition in the 2-D case by utilizing a 4-channel analysis filter bank first temporally and second spatially. The design and properties of non-unitary filter banks depend on the targeted applications. An overview is given in the next subsection. Please refer to Ramstad et al. (1995) and Balasingham (1998) for detailed issues on the design of non-unitary filter banks.

1.3.1 Application to seismic data

As previously mentioned, filter banks have been applied for instance to denoising of seismic data, to migration of seismic data, and to seismic data compression. Of these three applications the greatest success has without question been in the case of compression.

The main objective of the analysis filter bank is to decorrelate the input signal and compact the energy of the signal. Unfortunately, the analysis filter bank generally introduces aliasing, amplitude, and phase distortions which must be canceled by the synthesis filter bank in the PR case. In addition the M separate analysis-synthesis-filter pairs are not *linear time-invariant* systems due to the down-sampling and up-sampling procedures (see e.g., Villasenor et al., 1995); the output signals of the synthesis filter bank in channel or branch

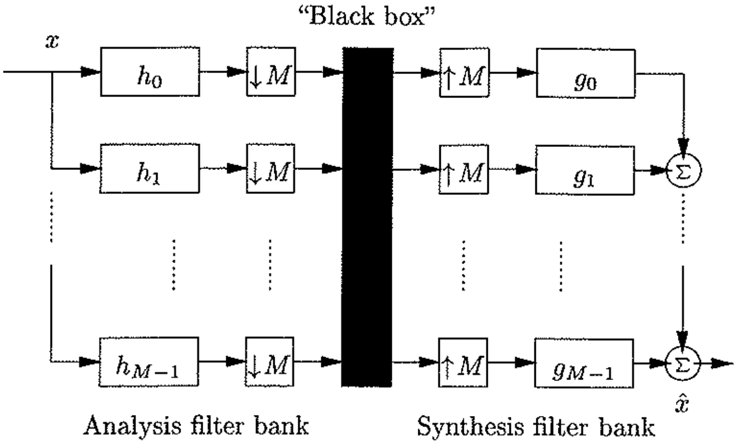


Figure 1.10: An M-channel parallel-structured uniform filter bank with a "black box" between the analysis and synthesis sides.

number m from an input signal and a delayed version of the same input signal are not simply delayed versions of each other, but they can be highly different. This can cause some problems at least in the case of seismic migration.

For all applications a "black box" is put between the analysis and synthesis filter banks (see Figure 1.10). In the case of denoising the "black box" is (soft or hard) thresholding of the subband samples (Strang and Nguyen, 1996; Duval and Røsten, 2000), and the analysis filters should optimally provide a good separation between signal and noise. In the case of migration the "black box" is composed of the wave-field extrapolation and imaging steps used in conventional (time-space domain) migration, but mapped to the subband domain. Unfortunately, unwanted cross-terms between all subbands are introduced due to the aliasing distortion introduced by the analysis filter bank (Røsten and Ramstad, 1998). In other words, such a scheme needs wave-field extrapolation filters from each subband to all the other subbands. As a consequence, migration in a subband domain provides a trade-off between resolution and efficiency. The scheme is either efficient giving low resolution (i.e., insufficient quality) or the scheme is slow having the same quality as provided by conventional migration techniques. The efficiency can be increased somewhat, while securing the quality, by well designed analysis filters; a high number of filter taps or coefficients (i.e., long impulse responses) can reduce the aliasing distortion. A big dilemma is, however, that the sparseness of the extrapolation filters at the same time is given by the lengths of the impulse responses of the analysis and synthesis filter banks. A better solution to diminish the aliasing problem is probably to perform a minor over-sampling, e.g., by a factor of $M/4$, in every subband channel.

In the case of SBC, the "black box" can be made up of uniform scalar quantization and entropy coding at the encoder side and of entropy decoding and inverse uniform scalar quantization at the decoder side. Now, the filter banks should be optimized by maximizing the so-called coding gain (Ramstad et al., 1995). SBC of seismic data is, nevertheless, not straightforward. Speech, audio, image, and video coding have SBC applications for storage and transmission, and the compression ratio may be set as high as desired, consistent with demands put forward by human perception. If further data processing is required, the processing is most conveniently carried out with the original uncompressed data. Compression of seismic data, on the other hand, is more complicated. For example, if compression is applied in the acquisition phase offshore, and the compressed seismic data set is transmitted to land, the effects of compression noise on the seismic data must be negligible since the complete processing flow is carried out with the seismic data after decompression. This very important complication is to be discussed in this dissertation.

1.4 Directions for future research

The use of seismic data compression raises several questions. Naturally, many of these will not be answered completely in this dissertation. Some suggestions for future research within the area of seismic data compression using SBC are for instance:

- compression of other prestack seismic data sets, both land and marine;
- extensive subjective evaluation of the reconstructed seismic data performed by non-biased geoscientists, for example side by side comparisons of original and decompressed prestack and poststack seismic data;
- development of safe SBC algorithms for seismic data based on for example the level and characteristics of ambient noise components (see e.g., Donoho et al., 1999b, 2000);
- SBC methods for seismic data using fast and flexible implementations of filter banks such as for example provided by the so-called lifting schemes (see e.g., Claypoole and Baraniuk, 1998; Khène and Abdul-Jauwad, 2000).

1.5 Organization

The remainder of the thesis is composed of three papers on seismic data compression submitted to an international journal where the papers are currently undergoing the review process. Chapter 2 (Paper 1), Chapter 3 (Paper 2) and Chapter 4 (Paper 3) are written as self-contained papers, with their own abstract, introduction, discussion, conclusions, and appendices. Consequently, redundancy between the material covered in each chapter is difficult to avoid, but the overlaps are kept to a minimum. The first paper deals with lossy compression of common offset gathers using SBC, the second paper considers lossless coding of the associated trace identification headers using entropy coding, and the third paper analyzes the influence of lossy seismic data compression on processing and interpretation. Following the papers, a complete references list is presented. An index is given at the end of the thesis.

The three main chapters can be read separately in any preferred order though it is natural to read them in chronological order. For all papers, valuable contributions have been given by the two supervisors in addition to one former and one present colleague of the undersigned. The titles of the papers and the names of the co-authors with corresponding affiliations are listed below:

Paper 1

Seismic data compression,
Part I: Subband coding of common offset gathers
Tage Røsten[*], Tor A. Ramstad[*] and Lasse Amundsen[‡]

Paper 2

Seismic data compression,
Part II: Lossless coding of trace identification headers
Tage Røsten[*], Tor A. Ramstad[*] and Lasse Amundsen[‡]

Paper 3

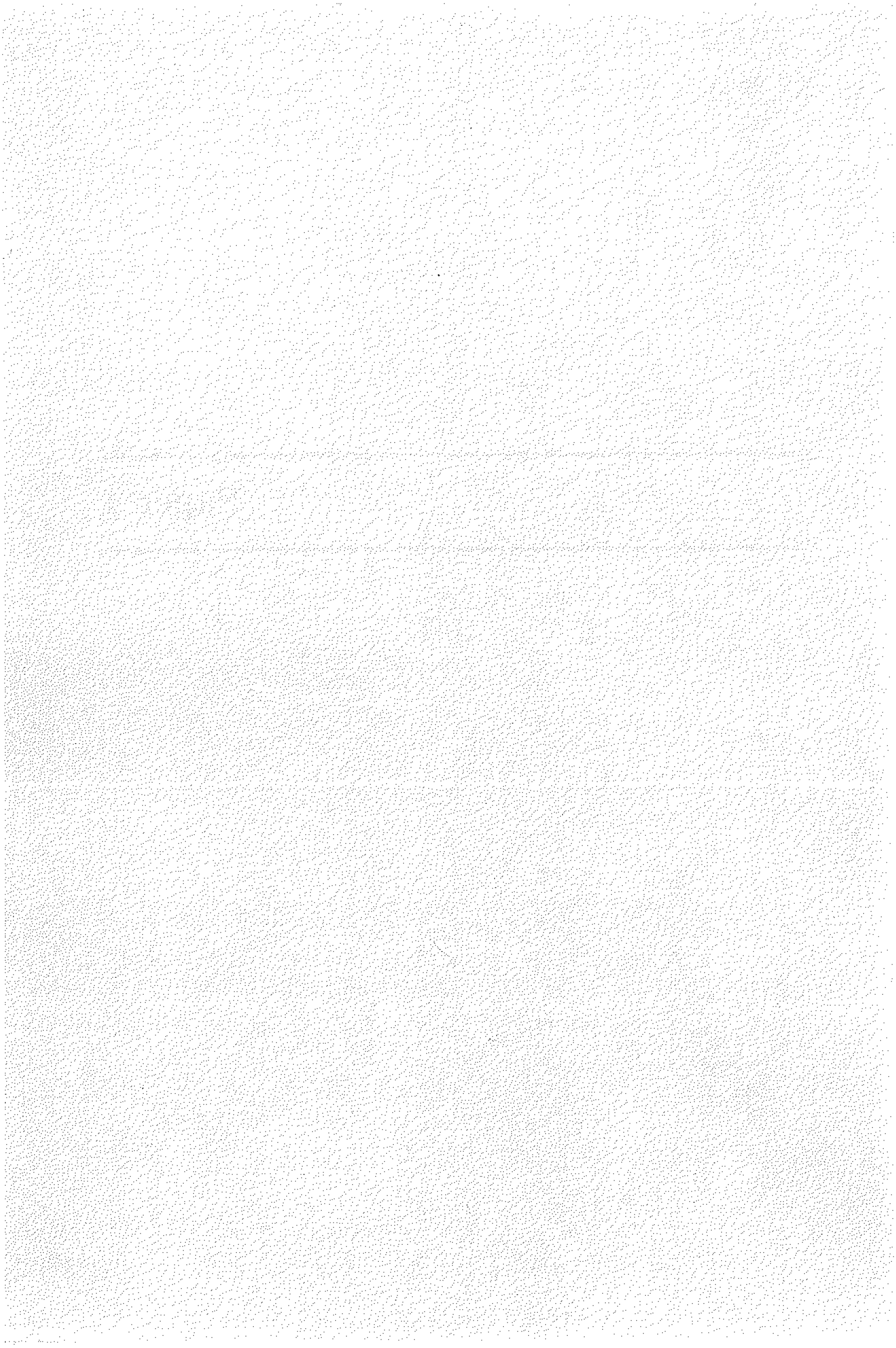
Seismic data compression,
Part III: Its influence on processing and interpretation
Tage Røsten[*], Patrick Waldemar[‡,§], Arild Buland[‡] and Lasse Amundsen[‡]

[*] Norwegian University of Science and Technology.

[‡] Statoil Research Centre.

[§] Now: Fast Search & Transfer.

Paper 1



Chapter 2

Seismic data compression, Part I: Subband coding of common offset gathers

2.1 Abstract

Seismic data require huge transmission capacities and large storage mediums, and the volume of seismic data is continuing to increase rapidly due to, e.g., acquisition of large 3-D surveys, re-processing of prestack seismic data, and calculation of poststack seismic data attributes. We consider lossy compression as an important tool for efficient handling of large seismic data sets. We present a 2-D lossy seismic data compression algorithm based on subband coding, and focus on adaptation and optimization of the method for common offset gathers. The subband coding algorithm consists of five stages: First, a preprocessing phase using an automatic gain control to decrease the non-stationary behavior of seismic data. Second, a decorrelation stage using a uniform analysis filter bank to concentrate the energy of seismic data into a minimum number of subbands. Then, an iterative classification algorithm based on the variance to blocks of subband samples to classify the subband samples into a fixed number of classes with approximately the same statistics. Fourth, a quantization step using a uniform scalar quantizer which gives an approximation of the subband samples to allow for high compression ratios. And finally, an entropy coding stage using a fixed number of arithmetic encoders matched to the corresponding statistics of the classified and quantized subband samples to actually achieve compression. Decompression basically performs the opposite opera-

tions in reverse order. We compare the proposed subband coding algorithm with three other seismic data compression algorithms. The high performance of our subband coding scheme is supported by objective and subjective results. For the common offset gather example, a compression ratio of 10:1 gives compression noise below the level of the ambient noise, while the compression noise is hardly visible at a compression ratio of 30:1.

2.2 Introduction

The term *data compression* refers to the process of reducing the amount of data (number of bits per sample) required to represent a given quantity of information of the signal. A clear distinction must be made between data and information. They are not synonymous. In fact, data are the means by which information is conveyed. Various amounts of data may be used to represent the same amount of information. Some data provide no relevant information or simply restate already known information. Data are thus said to contain redundancy. In addition, some data simply have less relevant information than other data, and thus data are said to contain irrelevancy. On the contrary, the amount of relevant information inherent in the data typically change from application to application. In other words, data with relevant information for one application can be data with irrelevant information for another application. We thus say that the two applications have different tolerance levels for the information. Data compression is achieved when redundancy is reduced, irrelevancy is eliminated, and/or an accepted tolerance level for the information is reached. When the redundancy is reduced from the data, while the irrelevancy is not utilized, data compression will not introduce distortion. This approach is called *lossless data compression*. On the other hand, when redundancy and irrelevancy are exploited, distortion is introduced. This approach is called *lossy data compression*. Due to the introduction of compression distortion, lossy methods generally offer smaller number of bits per sample, i.e., lower bit rates or higher compression ratios, than lossless methods. As a rule of thumb, a compression ratio greater than 2:1 (for floating-point data) requires a lossy method. Compression of text, on the other hand, can achieve significantly higher compression ratios than 2:1 by lossless methods. This is because a text file contains frequently repeated letters and patterns of letters while a typical seismic data file is far less structured.

When we are talking about a compression system we are actually referring to two algorithms: The compression algorithm at the encoder side generating a compressed signal, and the decompression algorithm at the decoder side generating a reconstructed signal. We follow the convention of referring to both the compression and decompression parts of the system as the compression algorithm.

The basic problem in lossy data compression is to achieve the minimum possible distortion for a given *bit rate*, or equivalently, to achieve a given acceptable level of distortion with the least possible bit rate (Jayant and Noll, 1984). This is described by the rate-distortion function. In the rest of this paper, the bit rate is specified as the number of bits per sample. The distortion can be measured by *objective* measurements such as signal-to-noise ratio and mean-square error, or by *subjective* (e.g., visual) measurements. An additional parameter that enters the picture is complexity of compression, but the role of this is commonly ignored.

Bit rate is commonly classified into low and high. For simplicity we say that bit rates lower than "approximately" 3.2 bits per sample are low bit rates, while bit rates higher than 3.2 bits per sample are high bit rates. Seismic data are most frequently represented by 32 bits per sample (floating-point number representation). Thus, a compression ratio of 16:1 makes the "limit" between low and high compression ratios of seismic data.

2.2.1 Lossy compression of seismic data

Lossy compression is the tool for efficient transmission and storage of prestack seismic data. We claim that the distortion introduced by lossy compression can be regarded as harmless for seismic data processing and interpretation if the compression noise has an amplitude level below, and has similar characteristics (e.g., white) as ambient noise components. Our claim is supported by the literature. Seismic data compression is not dangerous to processes that attenuate noise in the seismic data, for example migration and stacking (see e.g., Bosman and Reiter, 1993; Hall et al., 1995; Bradley et al., 1996; Røsten et al., 2000b). Migration and stacking are beneficial procedures after decompression for compression ratios up to at least 20:1. Deconvolution, on the other hand, can be a harmful process after decompression already at a compression ratio of 10:1 since preserved phase appears to be crucial for deconvolution (Guo and Burrus, 1996). However, reducing the dynamic range of the seismic data by reversible amplitude balancing and emphasis filtering can somewhat circumvent this problem (Polzer et al., 1997).

To the authors' knowledge, the first significant article in the area of seismic data compression was written by Wood (1974). He presented a transform coding (TC) method based on the discrete Walsh-Hadamard transform (DWHT; Jain, 1989). Although the quality of the reconstructed seismic data set was satisfactory for visual inspection, which was Wood's dominant concern, its limited quality tended to degrade the performance of more complicated seismic data interpretation tools. The basis functions of the DWHT are orthogonal binary sequences of amplitudes plus and minus one, and this transform is suitable for the compression of data with sharp discontinuities, but not to compress seismic data which consist of oscillating functions. A study on seismic data compression using TC was presented by Spanias et al. (1991). More recently, Waldemar et al. (1997) proposed a TC technique of seismic data using the singular value decomposition (SVD; Jain, 1989). Vermeer et al. (1996), Duval et al. (1999a,b), and Wang and Wu (1999) have presented variants of more traditional block oriented TC methods of seismic data compression by designing smoothed and overlapped (trigonometric) analysis-synthesis basis functions, a concept first introduced by Malvar and Staelin (1989).

Previous work on lossy compression of seismic data using two related techniques to TC, that is discrete wavelet transform coding (DWTC) and subband coding (SBC), shows that the data organization (e.g., dimensionality and sorting) is important to achieve large compression ratios with an acceptable distortion level (Villasenor et al., 1996; Røsten et al., 1999a). Luo and Schuster (1992) used a DWTC based technique to compress seismic traces individually, but this 1-D approach limits the potential compression ratio to approximately 5:1 since the high trace-to-trace spatial correlation is not exploited (see also Chen, 1995). Using a 2-D data organization, compression ratios of 100:1 (Røsten et al., 1996) for poststack seismic data and 20:1 (Reiter and Heller, 1994) for prestack seismic data are achieved with small degradation. Nevertheless, in the case of poststack amplitude analysis, prestack seismic data can probably tolerate higher compression ratios than poststack seismic data due to the averaging along stacking trajectories (Reiter and Heller, 1994). With 3-D and 4-D data organizations compression ratios of 50:1 and 100:1, respectively, are possibly achieved for prestack seismic data with small resulting distortion. In this context, high-dimensional data organization does not refer to 3-D or 4-D data surveys in general. High-dimensional data organization rather means data sets structured into one temporal and at least two spatial dimensions. A quantitative comparison of 1-D, 2-D, and 3-D DWTC of seismic data has been done by Reiter (1996). A full utilization of the dimensionality makes the seismic data compression easier due to increased redundancy. Notwithstanding, it becomes more complicated to restore for instance single traces after high-dimensional compression since large volumes of traces must be decompressed. To circumvent this problem, Donoho et al. (1998) simply suggested to divide the seismic data set into overlapping (to prevent blocking artifacts) sub-volumes with separate compression of each sub-volume. Such division into stationary sub-volumes is in any case necessary for large seismic data sets.

In this paper, we consider compression of 2-D prestack seismic data acquired offshore. Notice, compression of seismic data acquired on land is usually more involved than compression of marine seismic data due to, e.g., overwhelming amounts of non-white noise such as ground roll (Ergas et al., 1996). We focus on the optimization of a 2-D SBC scheme using a proper organization of the seismic data, i.e., common offset gathers. However, the proposed SBC algorithm can be easily extended to an arbitrary number of dimensions and to other domains of seismic data, for example common-mid point gathers.

This paper is organized as follows. First, we discuss some important characteristics of seismic data which put some constraints on the compression of seismic data. Second, we present the principles of SBC based on a uniform analysis filter bank, a uniform scalar quantizer, and arithmetic coding. We stress the strong relationship between SBC, DWTC, and TC. Next, the proposed SBC method is adapted to seismic data, and the optimization procedure is explained in detail. In the results section, we compare our SBC method with three other seismic data compression algorithms (Duval et al., 1999b; Donoho et al., 1995; Chen, 1995). These results are based on comparisons of the data immediately after decompression. The high performance of the proposed SBC algorithm is supported by objective and subjective results of a common offset gather example. Finally, discussion and conclusions end the paper.

This article is the first in a collection of three papers on seismic data compression. Lossless compression of the trace identification headers of the SEG-Y standard based on entropy coding is presented in Part II (Røsten et al., 1999c) while an analysis of the influence of lossy seismic data compression on processing and interpretation, using the proposed SBC, is presented in Part III (Røsten et al., 2000b).

2.3 Important characteristics of seismic data

Seismic data have several characteristics which have serious implications for compression. Seismic data must always be processed. After processing, seismic data are interpreted.

The most important attribute with respect to compression is that the signal energy is mostly concentrated at low frequencies and low wavenumbers of the seismic data. Hence, the information content in the seismic data is low. The seismic data can therefore be compressed substantially using lossy methods. Moreover, seismic data are frequently over-sampled in time which gives redundancy in the temporal direction of seismic data. Nevertheless, the redundancy in raw common shot gathers is not particularly high neither vertically nor horizontally. In SBC, a 1-D analysis filter bank is commonly applied separately along the temporal and spatial directions. Thus, the signal decomposition does not exploit the coherency or correlation of the raw common shot gathers. Exploiting coherency motivates the application of a non-separable 2-D filter bank. Also, a different organization or domain of seismic data with better coherency might be tried.

For instance, we can alternatively compress common mid-point gathers, which can be obtained from sorting the common shot gathers almost in real time, and apply a normal move-out (NMO) correction to horizontally align the primary reflection events. This approach has been suggested by for instance Reiter and Heller (1994) in the case of seismic data compression for the purpose of quality control. However, a brute NMO correction gives only minor improvements in the compression results (Røsten and Waldemar, 1998). A well adapted NMO correction requires a high number of velocities which must be sent as side information from the encoder to the decoder.

A more convenient organization of seismic data, at least in the case of seismic data compression for the purpose of storage, seems to be common offset gathers. Compression of common offset gathers is appropriate for three main reasons: First, the primary reflection events will be horizontally aligned for a non-dipping sub-surface. Second, a common shot gather or a common mid-point gather are eventually reconstructed from a set of independently decoded common offset gathers. The distortion introduced in the reconstructed common shot gathers or common mid-point gathers will therefore, intuitively, be less correlated. This second property is important because the majority of seismic data processing techniques are utilized on common shot gathers or common mid-point gathers. Prestack migration, on the other hand, is commonly applied on common offset gathers. Third, the direct wave and the sea-floor reflection (at small offsets) and the P-wave refractions (at large offsets) are easily detected on common offset gathers. The main part of the two-way traveltime (TWT) delay of the water layer above the first-break arrival can be efficiently compressed using, e.g., run-length coding (RLC; Nelson and Gailly, 1996).

Some of the advantages with using common offset gathers rather than common shot gathers in seismic data compression are demonstrated in Dessing and Hoekstra (1997) and Røsten et al. (1999a). Note, however, that the spacing between the traces in a common offset gather is usually higher than in a common shot gather. Therefore, in a steeply dipping environment, common offset gathers may be aliased and the coherency property will be lacking.

Unfortunately, the local statistics of seismic data are highly varying. Adaptation to this "non-stationary" behavior can be achieved by dividing the seismic data into more stationary sub-signals with separate compression of each sub-signal. A more straightforward approach to decrease the non-stationary response is to perform an amplitude balancing at the encoder (see e.g., Vermeer et al., 1996) combined with, in the case of subsequent true-amplitude seismic data processing, an inverse amplitude balancing at the decoder. This amplitude balancing should of course be adaptive to the seismic data, for example by automatic gain control (AGC), while at the same time the inverse gain function must be efficiently parameterized (compressed). Be aware that an amplitude balancing can simultaneously increase the magnitude of the ambient noise. After AGC and inverse AGC, a more uniform quality of the reconstructed seismic data is obtained at least in the deep zone. The number of bits used in the compression increases at the same time, directly and indirectly, due to the preprocessing. For example, after AGC, the deep zone of a common offset gather contains mostly seismic data with low coherency.

2.3.1 Preprocessing of seismic data

To maximize the compression performance, we have to carry out preprocessing of the seismic data. At the encoder side, we sort the seismic data from common shot gathers (CSGs) to common offset gathers (COGs) to match the signal decomposition to the correlation. For each COG, we compress the main part of the TWT delay of the water layer using RLC, i.e., the run of small amplitude values in the TWT delay of the water layer is set to zero and solely expressed by a count which represents the number of zeros in the run. Figure 2.1 shows an example of a COG without the main part of the TWT delay of the water layer. This COG is identical to the COG with offset equal to -2108.5 (or receiver number h = 161) of the data set used in Røsten et al. (2000b). The

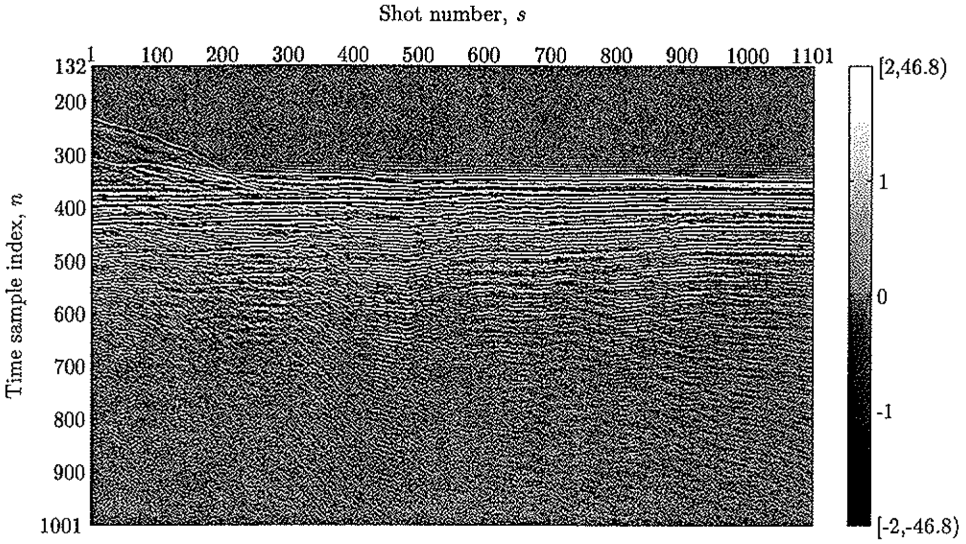


Figure 2.1: An example of a COG without the TWT delay of the water layer. The size is $(1001 - 132 + 1) \times 1101 = 870 \times 1101$ samples. The first time sample index and the first shot number start at 1. The TWT delay is modeled as the area from and above a horizontal line placed at time sample index $n = 131$.

TWT delay of the water layer is for simplicity modeled as the area in the COG from and above a horizontal line placed at time sample index $n = 131$. This horizontal line is placed in a secure distance from the direct wave and the P-wave refractions to prevent the appearance of edge effects to propagate into important areas of the seismic data. As the final step in preprocessing, we apply a 2-D AGC to decrease the non-stationary behavior of the COG.

The AGC is efficiently implemented by calculating the mean of absolute values of square blocks of $B \times B = 32 \times 32$ (i.e., height × width) adjacent non-overlapping samples. Smaller and larger block sizes, e.g., 16×16 and 64×64, give lower efficiency and resolution, respectively, of the estimated mean of absolute values. Since the coherency of a COG is larger in the lateral direction, a rectangular block size, e.g., 16×64, can be used instead.

The estimated gain values are compressed using a fixed-rate open-loop differential pulse code modulation (DPCM; Gersho and Gray, 1991). This method exploits the large lateral coherency of the gain values in a COG, and the details are: The gain values are scanned row-wise from left to right (to exploit the large lateral coherency) and column-wise from top to bottom. A simple 1-D predictor, using a first-order finite impulse response (FIR) filter with prediction coefficient equal to 0.50, and a mid-tread Lloyd-Max threshold scalar

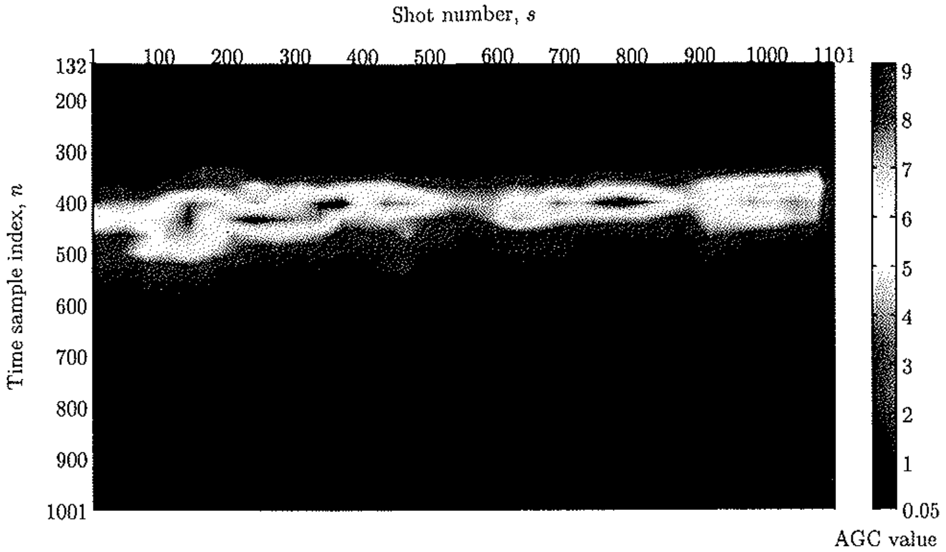


Figure 2.2: The AGC function derived from the COG shown in Figure 2.1. Dark blue color corresponds to 0.05, i.e., the utilized threshold value in the AGC scheme.

quantizer (see Gersho and Gray, 1991, and Figure 2.7) with 4 bits per sample (i.e., $2^4 = 16$ decision levels) are utilized in the DPCM scheme. The mean and standard deviation values from the DPCM scheme are sent as side information from the encoder to the decoder. We assume a Laplacian probability density function for the difference samples. The size of the file with compressed gain values is approximately 0.01 % of the size of the associated COG.

To obtain a perfectly reversible AGC the encoder and decoder sides must have identical replicates of the compressed mean of absolute values. To achieve this we decode the compressed gain values at both the encoder and decoder. Piecewise-linear interpolation between the decoded mean of absolute values is used separately in the temporal and the spatial directions to give a complete AGC function. AGC values below 0.05 are thresholded to prevent the domination of ambient noise with large magnitudes. Figure 2.2 displays the complete AGC function derived from the COG shown in Figure 2.1. At the encoder, the COG is divided sample by sample by this AGC function. Notice the reduced dynamic range obtained after applying the AGC (see Figure 2.3). At the decoder, on the other hand, the reconstructed COG is multiplied sample by sample by the same AGC function.

Applying some preprocessing procedures which can strengthen the coherency of the original seismic data is undoubtedly beneficial to the final result.

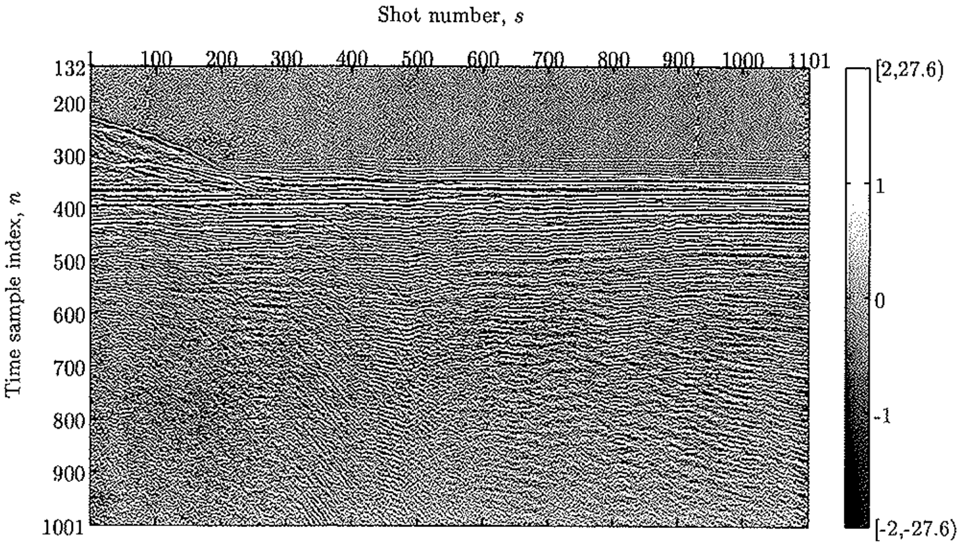


Figure 2.3: The COG shown in Figure 2.1 after AGC has been applied.

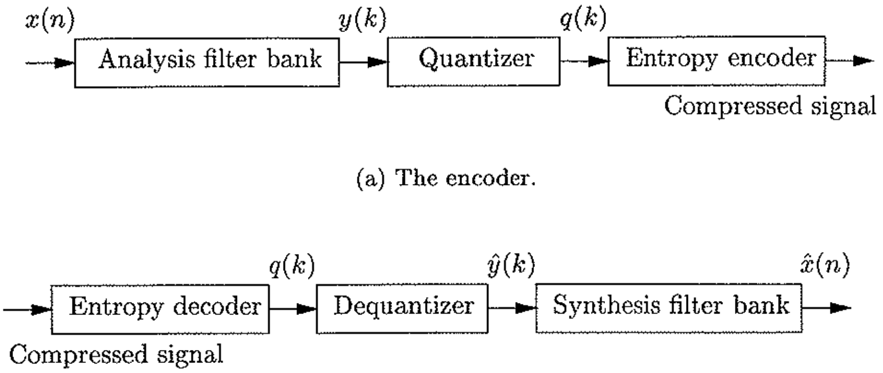
But there is a trade-off between the additional number of bits required and the time spent by preprocessing, and the effect achieved by such a procedure.

2.4 Subband coding – Introduction to our method

It follows from Shannon's data compression theorem that the best compression system can be designed by using *vector quantization* (VQ; Gersho and Gray, 1991) of the samples to the input signal. In theory, this design approach requires a compression system with infinite complexity and delay. In practice, further preprocessing of the input signal prior to quantization is performed to remove the statistical dependencies, e.g., redundancy or correlation, between the signal samples. Then, in the extreme case where all statistical dependencies are removed, simple *scalar quantization* (SQ), i.e., VQ with the number of dimensions equal to one, can be efficiently used (Jayant and Noll, 1984).

SBC has become quite popular in speech, audio, image, and video coding. A typical subband encoder [see Figure 2.4 (a)] consists of three parts: an analysis filter bank, a quantizer, and an entropy encoder. The subband decoder [see Figure 2.4 (b)] performs in general the opposite operations in reverse order: an entropy decoder, an inverse quantizer ("dequantizer"), and a synthesis filter bank.

The principles of the SBC method are: At the encoder side, the input sig-



(a) The encoder.

(b) The decoder.

Figure 2.4: A typical lossy SBC system. Of the six processes only the quantizer-dequantizer pair is lossy, the other pairs are reversible.

nal is decomposed into M spectral subbands by a uniform analysis filter bank. This subband decomposition decorrelates the input signal and concentrates the energy of the input signal into a minimum number of subbands. Then the subband samples in every subband are critically down-sampled by M (every Mth subband sample is retained) to keep the total number of subband samples unchanged compared to the original input signal representation. The down-sampled subband samples are quantized by a uniform scalar quantizer to eliminate the irrelevancy. The uniform scalar quantizer gives an approximation of the subband samples, and solely introduces the compression noise.[2.1] Finally, the quantized subband samples are entropy coded to achieve compression. At the decoder side, the M spectral subbands are entropy decoded and dequantized. Then the subband samples in every subband are up-sampled by M (that is inserting $M - 1$ zeros between every dequantized subband sample). At the end, the signal is reconstructed by a uniform synthesis filter bank. The only distortion introduced by the SBC method is the compression noise introduced due to the quantization process.

In this article, we use a lossy SBC method applicable for common offset gathers. The filter coefficients of the analysis and synthesis filter banks are adapted to the temporal and spatial correlations of COGs (explained later). The rest of the SBC system contains of conventional blocks (see Figure 2.4)

[2.1] The filter bank system is assumed to be perfect reconstruction.

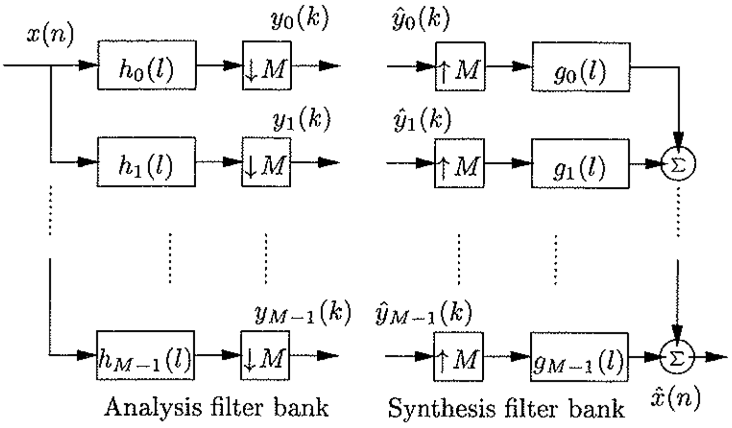


Figure 2.5: An M-channel maximally decimated filter bank system in the absence of quantization and entropy coding. The symbols $\downarrow M$ and $\uparrow M$ denote down-sampling and up-sampling by M, respectively.

except for (see Figures 2.12 and 2.13) the preprocessing of the COGs (using RLC/RLC^{-1} of the TWT delay of the water layer and AGC/AGC^{-1} of the COGs) and the classification of the subband samples into C number of classes (also explained later). We thus use C entropy encoders and decoders.

2.4.1 Filter bank signal decomposition and reconstruction

We assign the following capital letters:

N	–	the length or number of samples of a 1-D input signal;
M	–	the number of analysis and synthesis subband channels;
L	–	the number of filter taps of the analysis and synthesis filters;
K	–	the number of subband samples in each subband.

The subband decomposition and reconstruction system of COGs is based on separable parallel-structured uniform FIR analysis and synthesis filter banks. Figure 2.5 shows an M-channel parallel-structured uniform filter bank in the absence of quantization and entropy coding. The 1-D input signal $x(n)$ for $n = 0, 1, \ldots, N-1$ is decomposed into M subbands, each subband having K subband samples, by the uniform analysis filter bank. The input signal can be perfectly reconstructed from the M subbands in the absence of quantization (except from a pure positive delay) if the analysis and synthesis filters obey a so-called perfect reconstruction (PR) condition. Then

$$\hat{x}(n) = x(n - n_0), \tag{2.1}$$

where $\hat{x}(n)$ stands for the reconstructed signal and n_0 the delay.

We denote the analysis filters by $h_m(l)$ for $m = 0, 1, \ldots, M-1$ and $l = 0, 1, \ldots, L-1$, the subband signals by $y_m(k)$ for $k \in \mathbb{N}$, the reconstructed subband signals by $\hat{y}_m(k)$, and, finally, the synthesis filters by $g_m(l)$. The equations for the analysis and synthesis filter banks are (Ramstad et al., 1995)

$$y_m(k) = \sum_{n=-\infty}^{\infty} h_m(kM - n)x(n) \tag{2.2a}$$

and

$$\hat{x}(n) = \sum_{m=0}^{M-1} \sum_{k=-\infty}^{\infty} g_m(n - kM)\hat{y}_m(k), \tag{2.2b}$$

respectively.

The 2-D subband decomposition and reconstruction system for COGs is made by separately filtering, for instance, first row-wise and then column-wise at the analysis side, and vice versa at the synthesis side. The total number of subbands or subband images becomes $M \times M$ (see Figure 2.9 for $M \times M = 8 \times 8$).

The types of analysis and synthesis filters can be divided into three classes:

- orthogonal PR filter banks;
- biorthogonal PR filter banks;
- other non-unitary filter banks or filter banks without restrictions.[2.2]

Please refer to Strang and Nguyen (1996) for a precise definition of biorthogonality. For simplicity we say the analysis filters h_m for $m = 0, 1, \ldots, M-1$ are orthogonal to all reversed and shifted by kM, $k \in \mathbb{Z}$, versions of the synthesis filters g_m, but the h_m's and the g_m's are not orthogonal themselves.

Alias, amplitude, and phase distortions are introduced by the analysis filters, and the synthesis filters seek to cancel out these deformations. However, in the case of lossy compression, the analysis and synthesis filters need not to be

[2.2] Biorthogonal filter banks are strictly speaking also non-unitary, but non-unitary filter banks need not to be PR.

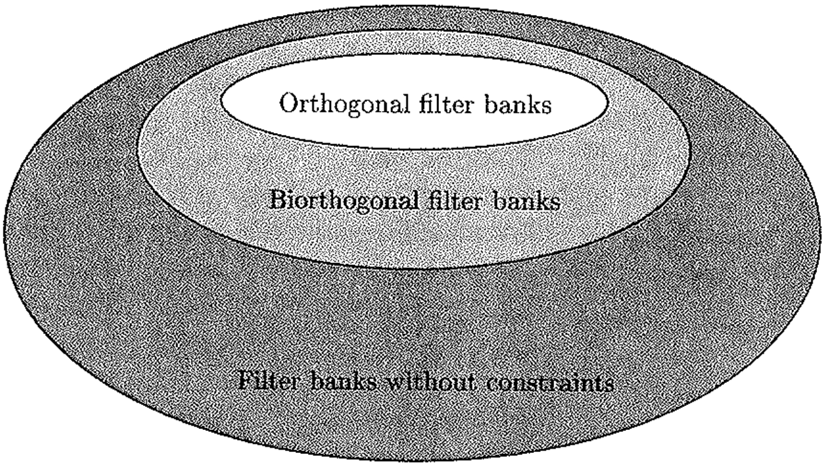


Figure 2.6: Venn-diagram of different sets of filter banks. Remark, orthogonal and biorthognal *wavelets* are subsets of orthogonal and biorthogonal filter banks, respectively.

a PR system since distortion in any case will be introduced by the quantization process. Actually, at low bit rates, the optimal filter bank in a mean-square error (mse) sense is far from PR (Hjørungnes et al., 1999). Hence, *orthogonal* and *biorthogonal* filter banks are not optimal to use in the mse sense since they are always PR (see also Figure 2.6). For high bit rates, on the other hand, the optimal filter bank in the mse sense is close to a PR system.

2.4.1.1 Maximally decimated filter banks

For maximum compression efficiency, we prefer a non-expansive subband decomposition to retain the length of the input signal, N, after subband reconstruction. N divided by M has to be equal to K, i.e., the number of subband samples in each subband, K, should ideally be the same for all subbands. This is accomplished by three steps: First, the last sample of the input signal is repeated until the required input signal length is reached. Second, intelligent extension of the edges of the input signal (see below) prior to the filter bank signal decomposition is performed. Third, the subband samples are critically down-sampled by M. The combination of these operations gives a maximally decimated filter bank system with the property $K \times M = N$. Moreover, both $x(n)$ and $\hat{x}(n)$ will have lengths equal to N. Note that the lower and upper summation indices for n and k in equations 2.2a and 2.2b are for simplicity set to $-\infty$ and ∞, respectively, due to the application of this three steps.

The main techniques for the second step are: zero, circular, and mirror extensions of the input signal edges (see e.g., Martucci, 1991; Brislawn, 1996). Zero and circular extensions generally introduce a discontinuity of the input signal at the edges. Mirror extension (used by e.g., Duval et al., 1999a,b) on the other hand, generally introduces a discontinuity in the first derivative of the input signal at the edges. In coping with edge effects, the mirror extension method, which requires linear-phase or symmetric filters (for example the most common 2-channel biorthogonal wavelets and M-channel for $M \neq 2$ orthogonal and non-unitary filter banks), offers better compression results compared to the zero (used by e.g., Wang and Wu, 1999) and circular extensions. Zero and circular extension methods are utilized in the case of non-symmetric filters (e.g., all non-trivial 2-channel orthogonal filter banks).

2.4.1.2 Frequency domain coders

TC and DWTC are strongly related to SBC. All three are based on filter banks and belong to a class of compression systems called frequency domain coders. The filter bank signal decomposition in TC [e.g., using the discrete cosine transform (DCT)] is strictly block oriented, i.e., $L = M$, whereas the subband decomposition in SBC allows also for overlapping blocks, i.e., $L > M$. Blocking effect and ringing noise are the artifacts experienced in traditional TC and SBC, respectively. SBC has benefited considerably from TC both in theory and practice (Ramstad et al., 1995). The DCT is the most well-known linear-phase orthogonal PR filter bank, and the DCT is currently used in the Joint Photographic Expert Group (JPEG) image compression standard. Other examples of linear-phase orthogonal PR filter banks are the lapped orthogonal transform (LOT; Malvar and Staelin, 1989) and the generalized LOT (GenLOT; Queiroz et al., 1996). The DCT and LOT are special cases of GenLOT with $L = M$ (i.e., no overlap) and $L = 2 \times M$, respectively.

The SBC was developed by signal processing researchers for speech coding by Crochiere et al. (1976) and for image compression by Woods and O'Neil (1986). The DWTC introduced some years later in applied mathematics has been recognized as a different view of SBC (see e.g., Vetterli and Herley, 1992). In DWTC, 2-channel orthogonal or biorthogonal wavelets (which are special cases of orthogonal and biorthogonal filter banks, respectively) are typically used in a tree-structured dyadic or octave-band filter bank. This type of filter bank is also sometimes called a logarithmic filter bank since the bandwidths of the subbands are equal on a logarithmic scale.

The orthogonality constraint is a very strong condition in 2-channel filter bank design. For instance, it is not possible to have both orthogonal PR and linear-phase filters except for trivial cases. The orthogonality constraint is relaxed for biorthogonal and other non-unitary filter banks. As a consequence, in the 2-channel case, biorthogonal filter banks may simultaneously be PR and have linear-phase – non-trivial – filter coefficients. The linear-phase property is commonly used to avoid for example artifacts such as edge effects in data compression. By relaxing the orthogonality or biorthogonality constraint even greater degrees of freedom will arise.

In the rest of the thesis, the main focus is on M-channel parallel-structured uniform filter banks using non-unitary linear-phase near-PR analysis and synthesis filters. Specifically, we use $M = 8$ number of subbands filter banks and the number of filter taps is $L = 32$. Please refer to Ramstad et al. (1995) for detailed issues on the design of non-unitary filter banks and SBC in general.

2.4.2 Quantization

The quantization process introduces the main loss in the SBC method. Consequently, careful design of the quantization procedure is extremely important in obtaining good compression results. In the case of seismic data, we are often interested in the large amplitudes to estimate various seismic parameters. We therefore want to keep these seismic data in position and amplitude as accurately as possible. A uniform scalar quantizer might in this case be a good choice for allocating the compression error in an optimal way for our purposes. At the same time, the small subband samples might often be classified as white ambient noise. A uniform scalar quantizer with dead-zone around zero out-passes some of the ambient noise components (see e.g., Chen, 1995). As pointed out by Farvardin and Modestino (1984), the *mid-tread* uniform scalar quantizer *with dead-zone*[2.3] (see Figure 2.7 and below for description) is close to the optimal entropy-coding constrained scalar quantizer.

Since we employ entropy coding, a mid-tread uniform threshold scalar quantizer (Gersho and Gray, 1991) of COGs after subband decomposition is therefore utilized. SQ can be viewed as a nonlinear mapping of $y_m(k) \equiv y(k) \in \mathbb{R}$ to a finite set $\gamma = \{\gamma(0), \gamma(1) \ldots, \gamma(I-1)\}$. The numbers $i = 0, 1, \ldots, I-1$ and the numbers $\gamma(i)$ are called the quantizer indices and the *quantizer representation levels*, respectively. The distance between the *quantizer decision levels* is termed the quantizer step-size and denoted Δ. The step-size is constant for a uniform scalar quantizer. Finally, $T = 2\beta \times \Delta$ for $\beta > 0$ represents

[2.3] For simplicity called a *mid-tread* uniform *threshold* scalar quantizer.

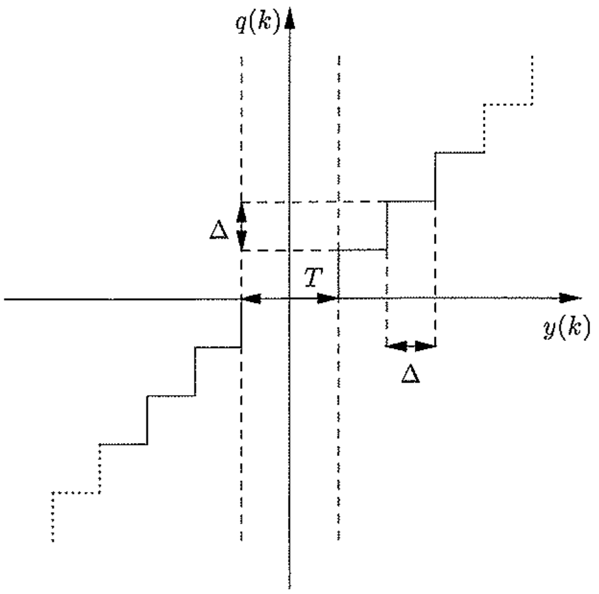


Figure 2.7: Mid-tread uniform threshold scalar quantizer with infinite range. The scalar quantizer is *mid-tread* since the origin is in the middle of a tread. The step-size is denoted Δ, and T represents the width of the dead-zone (i.e., the threshold) around zero.

the total width of dead-zone around zero. In image compression, $\beta = 0.5$ is commonly used (Ramstad et al., 1995) which provides no dead-zone. A dead-zone with $\beta = 0.6$ has been used for seismic data (Donoho et al., 1995; Villasenor et al., 1996), and $\beta = 0.6$ is also used in this work. Small compression ratios (i.e., lower than 10:1) requires $\beta < 0.5$ to give low amount of quantization noise.

We divide the quantization process into two operations, although the two steps are not reversible: a quantizer at the encoder and an inverse quantizer at the decoder. We denote the quantizer output by $q(k)$ (see Figure 2.7) and the reconstructed subband signal by $\hat{y}(k)$ [see Figure 2.4 (b)]. The quantizer operator, $q(k) = \mathrm{SQ}\{y(k)\}$ for $q(k) \in i$, selects the quantizer index according to (I is even):

$$i = \begin{cases} I/2 + \lfloor (y(k) + T/2)/\Delta \rfloor, & y(k) \leq -T/2 \\ I/2, & -T/2 < y(k) < T/2 \\ I/2 + \lceil (y(k) - T/2)/\Delta \rceil, & y(k) \geq T/2. \end{cases} \tag{2.3a}$$

The inverse quantizer operator, $\hat{y}(k) = \mathrm{SQ}^{-1}\{q(k)\}$ for $\hat{y}(k) \in \gamma(i)$, finds the quantizer representation level by

$$\gamma(i) = (i - I/2) \times \Delta. \tag{2.3b}$$

The quantizer representation levels are choosed to be equal to the *mid-points* of the corresponding quantizer decision intervals (see equation 2.3b) although the *centroids* are optimal to use in the mse sense (Gersho and Gray, 1991). Thus, the step-size will be given by $\Delta = |\gamma(j) - \gamma(j-1)|$ for $j = 1, 2, \ldots, I-1$.

Remark, the dynamic range of the mid-tread uniform threshold scalar quantizer is exceeded if

$$y(k) < -I/2 \times \Delta - T/2 \quad \text{or} \quad y(k) > (I/2 - 1) \times \Delta + T/2. \tag{2.4}$$

Then, equation (2.3a) must be exchanged with

$$i = 0 \quad \text{or} \quad i = I - 1, \tag{2.5}$$

respectively.

2.4.3 Entropy coding

The last step in the SBC system is reversible entropy or variable-rate coding of the quantizer outputs, going from a fixed number to a variable number (but a lower number in mean) of bits per quantizer index. The idea is to assign short bit-codes to quantizer indices that appear frequently while reserving long bit-codes to less frequent ones, thus achieving compression. Here, arithmetic coding (AC) of the quantizer outputs, representing the COGs after subband decomposition and quantization, is utilized. AC replaces the sequence of quantizer outputs (called a message) by a single number lying between two real numbers, P_b and P_e, such that $0.0 \leq P_b < P_e < 1.0$, representing a half-open interval $[P_b, P_e)$. More bits for the interval are needed in the resulting bit-code for longer messages. Each quantizer output added to the message incrementally modifies the bit-code. Thus, the net effect of each quantizer output on the resulting bit-code can be a fractional number of bits instead of an integral number of bits as in the case of Huffman coding (used by e.g., Donoho et al., 1995). AC typically offers higher compression ratios than Huffman coding (HC), but is more complex to implement (Witten et al., 1987). In both cases, a good estimate of the actual probability density function of the quantizer indices to be compressed is needed for efficient compression. A simple numerical example of entropy coding using AC and HC is given in Røsten et al. (1999c).

Consider the I possible quantizer indices, $i = 0, 1, \ldots, I-1$, and their associated probability of occurrences, $P_Q(i)$ (the quantizer indices are represented by a real stochastic variable Q), satisfying

$$\sum_{i=0}^{I-1} P_Q(i) = 1.0. \tag{2.6}$$

We want to arithmetic encode the sequence of quantizer outputs, $q(k) \in i$, of length K from all quantized subbands (see e.g., Nelson and Gailly, 1996):

1. initialize the so-called current interval to $[0.0, 1.0)$, that is $P_b = 0.0$ and $P_e = 1.0$;
2. for all $q(k)$, starting with the first and ending with the last quantized subband sample, do the following:
   (a) split the current interval to I subintervals, e.g., a subinterval for each of the possible quantizer indices. The width of the ith subinterval is proportional to its associated probability of occurrence, $P_Q(i)$;
   (b) the subinterval which corresponds to $q(k)$ is the next current interval, e.g., the jth subinterval is the next current interval if $q(k) = j$, $j \in i$.
3. Set $[P_b, P_e)$ equal to the two probabilities defining last current interval;
4. the arithmetic encoder allocates a number of bits for the sequence of quantizer outputs large enough to distinguish the last current interval from all possible last current intervals. The arithmetic decoder knows the length K of the sequence of quantizer outputs.

The lower bound of the average number of bits per sample needed for lossless compression of the quantizer indices, assuming independent and identically distributed (i.i.d.) quantizer indices, is given by the so-called Shannon or first-order entropy:

$$H = -\sum_{i=0}^{I-1} P_Q(i) \log_2[P_Q(i)] \tag{2.7}$$

in bits per sample (or bits per quantizer index). In theory, AC can achieve this lower bound H (Witten et al., 1987).

The price paid for the bit rate efficiency in entropy or variable-rate coding compared to fixed-rate coding (e.g., Lloyd-Max scalar quantizer with bit allocation) is, as indicated by the name, the variability in bit rate. In practice, we must tune the quantizer step-size and/or the threshold of the uniform scalar quantizer until the desired bit rate is achieved. Fortunately, this search can be done quickly since the rate-distortion function is a convex function (Jayant and Noll, 1984).

2.5 Optimization of the subband coding method

Efficient optimization of a compression system calls for good statistical models of the signal to be compressed. In our case, two statistical models are required. One for handling the statistics of common offset gathers to optimize the filter bank with respect to coding gain, and another statistical model of common offset gathers after subband decomposition to optimize the variable-rate coding part with respect to rate-distortion performance. For the first model, we use separable and real-valued Gauss-Markov processes fitted to the correlations of COGs. For the second model, on the other hand, we use a memoryless infinite Gaussian mixture distribution fitted to the histograms of COGs after subband decomposition.

The COG with the smallest (i.e., near) offset is often used as a first-order approximation of the resulting poststack seismic data. It is therefore reasonable to use similar statistical models to optimize subband coding algorithms for COGs and poststack seismic data. The optimization of a SBC method for poststack seismic data has been presented by Røsten et al. (1996, 1997a).

2.5.1 Optimization of the filter banks

The analysis filter bank is usually optimized to decorrelate the original signal such that the signal energy is maximized in a minimum number of subbands. This property is known as redundancy reduction or energy compaction, and is obtained by maximizing the so-called coding gain to SBC. Additional practical constraints can be placed on the type of splitting and on length of the filters.

The coding gain to SBC over pulse code modulation, G_{SBC}, assuming a uniform non-unitary PR M-channel L-tap filter bank is – in dB – given by (Katto and Yasuda, 1991)

$$G_{\mathrm{SBC}} = 10 \times \log_{10} \left(\left[\prod_{m=0}^{M-1} A_m \right]^{-1/M} \right), \tag{2.8}$$

provided that the synthesis filters, $g_m(l)$, have unit norm:

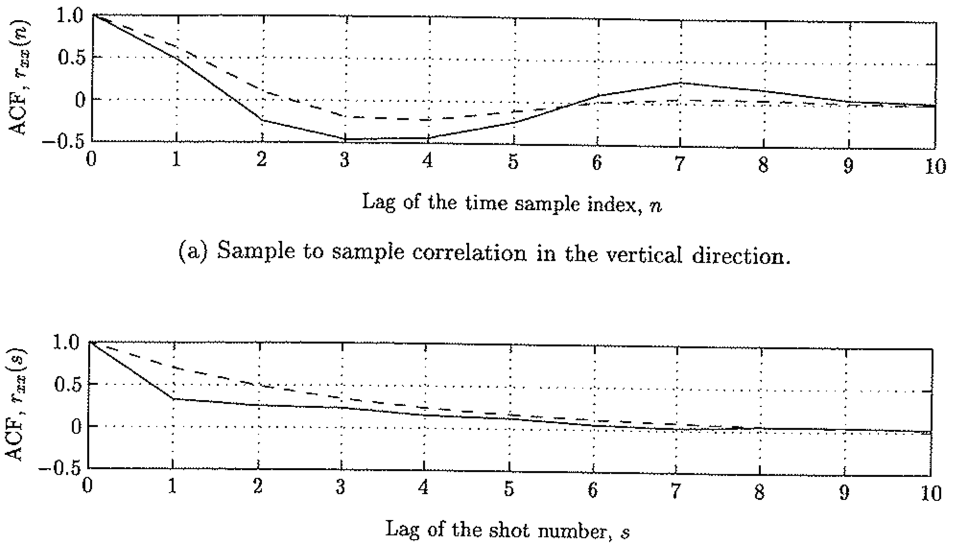
$$\sum_{l=0}^{L-1} g_m^2(l) = 1, \quad \forall m. \tag{2.9}$$

A_m in equation (2.8) can be calculated from the analysis filters, $h_m(l)$, and the autocorrelation function (ACF) of the 1-D input signal, $r_{xx}(n)$, in the following way

$$A_m = \sum_{j=0}^{L-1} \sum_{l=0}^{L-1} h_m(j) h_m(l) r_{xx}(j-l). \tag{2.10}$$

To find the filter banks optimized with respect to coding gain, we use the Nayebi method (see e.g., Nayebi et al., 1990) generalized by Aase (1993). Aase (1993) incorporated suitable error functions to guarantee the design of non-unitary linear-phase FIR filter banks having almost PR, good suppression of blocking effects, as well as high coding gain (see Appendix 2.B). From equation (2.10) it is evident that we need to know the ACF of the input signal to maximize equation (2.8). In the 2-D case, we have to investigate the sample to sample correlations both vertically and horizontally. Figures 2.8 (a) and (b) show the normalized measured ACFs in the temporal and spatial directions, respectively, of the COG after AGC displayed in Figure 2.3.

We choose a direction dependent separable statistical model with different zero-mean autoregressive (AR) processes to represent the ACFs in the two directions. The ACFs of first-order and second-order AR processes are given



(a) Sample to sample correlation in the vertical direction.

(b) Sample to sample correlation in the horizontal direction.

Figure 2.8: The normalized measured ACFs from the COG shown in Figure 2.3 (solid lines) and the ACFs of the utilized AR processes (dashed lines).

in Appendix 2.A. Røsten et al. (1999a) used a second-order Gauss-Markov or AR(2) process and a first-order Gauss-Markov or AR(1) process to model the correlations of COGs *without* AGC in the vertical and horizontal directions, respectively: the selected normalized correlation coefficients were equal to $\rho_1 = 0.85$ and $\rho_2 = 0.50$ for the AR(2) model, and $\rho_1 = 0.70$ for the AR(1) model. Similarly, Røsten et al. (1996, 1997a) used AR(2), with $\rho_1 = 0.62$ and $\rho_2 = 0.10$, and AR(1), with $\rho_1 = 0.97$, models to represent the correlations in the vertical and horizontal directions, respectively, of poststack seismic data.

In this article, we have to estimate the correlations of COGs *with* AGC. AGC will mainly increase the temporal resolution of COGs, stacking will do the same. Hence, we expect the correlation in the temporal direction of COGs with AGC to approach the vertical correlation of poststack seismic data. The resolution and correlation in the lateral direction of COGs will not be affected significantly by AGC. Thus, the optimized filter bank in the vertical direction of COGs with AGC is for simplicity chosen to be the same as the optimized filter bank in the temporal direction for poststack seismic data. The optimized filter bank in the horizontal direction of COGs with AGC, however, is the same as the optimized filter bank in the spatial direction of COGs without AGC. Figures 2.8 (a) and (b) in addition to qualitative experiments for several common offset gathers, indicate that the ACFs of the selected AR processes provide a reasonable fit to sample to sample correlations of COGs for small lags, that is $n, s \leq 10$. For comparison, Duval et al. (1999a,b) optimized a GenLOT with respect to coding gain for seismic data using an AR(1) process with $\rho_1 = 0.95$. A similar model is commonly used for images (Ramstad et al., 1995). Nevertheless, this model is not well matched to seismic data.

As previously mentioned, we set the number of subbands to $M = 8$ and the length of the filters to $L = 32$. We use the so-called filter bank configuration I which allows for full length filters in all channels (Aase, 1993). The same filter bank properties are utilized in subband decomposition of poststack seismic data (Røsten et al., 1996, 1997a). Duval et al. (1999b), however, utilized an 8-channel GenLOT with the number of filters taps, L, ranging from 40 to 48. Their experiments confirmed that longer, overlapping basis functions will give the best compression results (see also Vassiliou and Wickerhauser, 1997). We assign the following names for the selected filter banks: 8-32-I_062_010 for the filter bank in the time dimension and 8-32-I_070 for the filter bank in the space dimension. Tables 2.B-2 to 2.B-5 in Appendix 2.B quote the 8-channel 32-tap uniform filter banks.

Figure 2.9 presents the COG shown in Figure 2.3 after subband decomposition. The energy of this COG is mostly concentrated at subband number 1 or 2 temporally (i.e., bandpass) and at subband number 0 spatially (i.e., lowpass). Note that further subband decomposition of such high energy subbands with high remaining correlation can increase the compression performance.

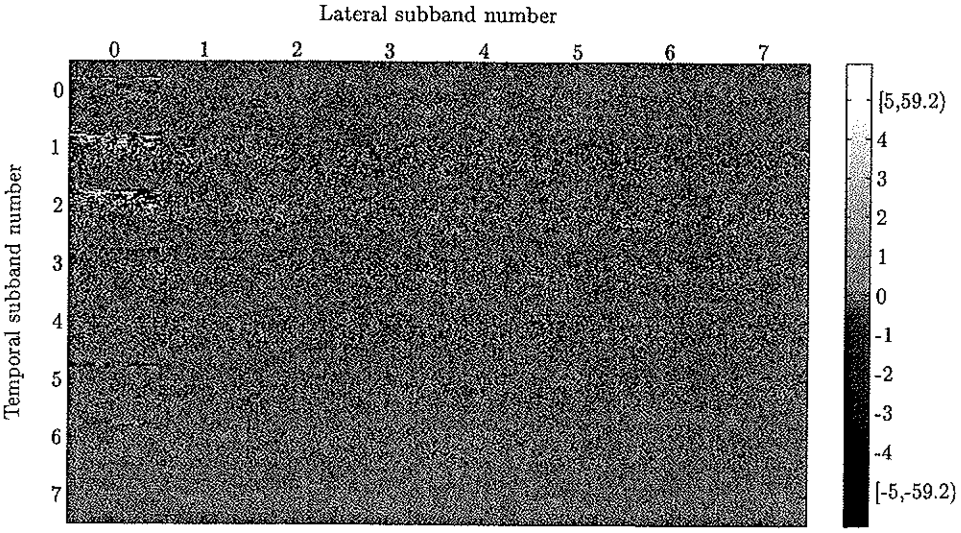


Figure 2.9: The COG shown in Figure 2.3 after subband decomposition into $M \times M = 8 \times 8$ number of subbands. The size is $(8 \times 109) \times (8 \times 138) = 872 \times 1104$ subband samples. The lowpass-lowpass subband (with numbers 0,0) is located in the upper-left corner. Clearly, the bandpass-lowpass subbands with numbers 1,0 or 2,0 pick up most energy. This is reasonable since COGs typically are bandpass and lowpass signals in the temporal and lateral directions, respectively.

2.5.2 Optimization of the variable-rate coders

An efficient coder has to adapt dynamically to the subband signal statistics. We use a method where the subband samples are classified into a fixed number of classes, each class consisting of subband samples with approximately the same statistics. After uniform SQ, each class is compressed class-wisely with a semi-adaptive arithmetic coder matched to the statistics of the class.

A uniform analysis filter bank is used for signal decomposition of COGs into $M \times M = 8 \times 8$ subbands, and square blocks consisting of $\tilde{B} \times \tilde{B}$ adjacent non-overlapping subband samples are classified into one out of C classes depending on the block energy, which is an asymptotically unbiased estimate of the variance (the so-called block variance) given that the blocks of $\tilde{B} \times \tilde{B}$ subband samples have zero-mean. A block size of $\tilde{B} \times \tilde{B} = 4 \times 4$ is found to give a satisfactory trade-off between local adaption and the amount of classification side information. As for the block size, the number of classes (arithmetic coders) is chosen as a compromise between the adaption and the amount of classification side information; $C = 5$ is found by experiments to give the best over-all compromise. Similar parameters are utilized in the proposed SBC method for poststack seismic data (Røsten et al., 1996, 1997a).

The next task is to model the statistics of the subband samples of common offset gathers. If we assume block-wise stationarity and statistical independent subband samples, the subband samples within each block will be i.i.d., and can therefore be characterized by a 1-D probability density function (pdf). The subband samples are represented by a zero-mean stochastic variable $Y \in \mathbb{R}$ with pdf $p_Y(y)$. Furthermore, we assume that a given block of subband samples is characterized by a single stochastic variable $\Sigma^2 \in [0, \infty)$, the variance, with pdf $p_{\Sigma_Y^2}(\sigma_y^2)$. Consequently, this results in a *memoryless infinite Gaussian mixture distribution* model (see also Lervik and Ramstad, 1996):

$$p_{Y|\Sigma_Y^2}(y|\sigma_y^2) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{y^2}{2\sigma_y^2}\right), \qquad (2.11)$$

where $p_{Y|\Sigma_Y^2}(y|\sigma_y^2)$ is a Gaussian pdf. The use of this model is justified by the fact that subband samples of seismic data are approximately zero-mean. In addition, subband samples of seismic data which have approximately the same estimated variance tend to have a Gaussian distribution (Lervik et al., 1996). Furthermore, the subband samples can be assumed to be uncorrelated (i.e., *memoryless*) and the number of estimated variances is typically very high (i.e., *infinite*).

We define $\tilde{\sigma}_{d,c}^2$ for $c = 0, 1, \ldots, C$ as the $C+1$ *variance decision levels* limiting the used range of each arithmetic coder. E.g., an estimated block variance of the subband samples, $\hat{\sigma}_y^2$, is classified into class c if $\hat{\sigma}_y^2 \in [\tilde{\sigma}_{d,c}^2, \tilde{\sigma}_{d,c+1}^2)$, where $\tilde{\sigma}_{d,0}^2 = 0$ and $\tilde{\sigma}_{d,C}^2 = \infty$. Likewise, we define $\tilde{\sigma}_{r,c}^2$ for $c = 0, 1, \ldots, C-1$ as the C *variance representation levels*. The variance representation levels specify the variances of the Gaussian probability density functions for which the C arithmetic coders could be optimized. However, instead of using Gaussian distributions with variances equal to the variance representation levels, better compression results are obtained by using the actual class histograms in the arithmetic coders (Lervik, 1996). Hence, the optimal variance representation levels can be interpreted as a tool for finding the optimal variance decision levels via an iterative algorithm to be described next.

The classification is adapted to each COG after subband decomposition. In a rate-distortion sense, assuming a Gaussian distribution, the optimal or maximum likelihood *variance decision levels* giving the optimal variance representation levels are (Lervik and Ramstad, 1996)

$$\tilde{\sigma}^2_{d,c,opt} = \frac{\ln\left(\dfrac{\tilde{\sigma}^2_{r,c,opt}}{\tilde{\sigma}^2_{r,c-1,opt}}\right)}{\dfrac{1}{\tilde{\sigma}^2_{r,c-1,opt}} - \dfrac{1}{\tilde{\sigma}^2_{r,c,opt}}}, \quad c = 1, 2, \dots, C-1. \tag{2.12}$$

On the contrary, the optimal or maximum likelihood *variance representation levels* giving the optimal variance decision levels are the *centroids* of the block variances belonging to the corresponding variance decision intervals. Thus, this optimization procedure is related to the optimization of Lloyd-Max scalar quantizers (Gersho and Gray, 1991). Although not strictly optimal, the variance representation levels are for simplicity set equal to the *mid-points* of the block variances. The iterative algorithm to find the optimal variance decision levels can be formulated as follows:

1. choose the block size $\tilde{B} \times \tilde{B}$, the number of classes C, and a stop-criterion $\epsilon > 0$;

2. set the bounding values of the optimal variance decision levels, that is $\tilde{\sigma}^2_{d,0,opt} = 0$ and $\tilde{\sigma}^2_{d,C,opt} = \infty$;

3. choose initial values for the rest of the variance decision levels $\tilde{\sigma}^2_{d,c,init}$ for $c = 1, 2, \dots, C-1$ where $0 < \tilde{\sigma}^2_{d,c,init} < \tilde{\sigma}^2_{d,c+1,init} < \infty$;

4. find the biased (maximum likelihood) estimates of the $\tilde{B} \times \tilde{B}$ block variances. The $\tilde{B} \times \tilde{B}$ blocks of subband samples are assumed to have zero-mean;

5. for $c = 0, 1, \dots, C-1$ find the optimal variance representation levels $\tilde{\sigma}^2_{r,c,opt}$ as the mid-points of the block variances belonging to the corresponding half-open initial variance decision interval $[\tilde{\sigma}^2_{d,c,init}, \tilde{\sigma}^2_{d,c+1,init})$;

6. for $c = 1, 2, \dots, C-1$ find the optimal variance decision levels $\tilde{\sigma}^2_{d,c,opt}$ using equation (2.12);

7. if $\sum_{c=1}^{C-1} |\tilde{\sigma}^2_{d,c,opt} - \tilde{\sigma}^2_{d,c,init}| > \epsilon$ then set $\tilde{\sigma}^2_{d,c,init} = \tilde{\sigma}^2_{d,c,opt}$ for $c = 1, 2, \dots, C-1$, and return to step 5. Else go to step 8;

8. stop the iterative algorithm.

Table 2.1: Optimal variance decision levels ($\tilde{\sigma}^2_{d,0,opt} = 0$ and $\tilde{\sigma}^2_{d,C,opt} = \infty$) and optimal variance representation levels. The block size is 4×4 and the number of classes $C = 5$.

	Block classification levels		Proportionality constants
	$\tilde{\sigma}^2_{d,c,opt}$	$\tilde{\sigma}^2_{r,c,opt}$	d_c
c			
0		0.01	
1	0.02	0.05	1.50
2	0.11	0.28	1.70
3	0.59	1.56	1.90
4	3.60	11.54	3.00

The optimal variance decision and representation levels of the COG after subband decomposition (see Figure 2.9) are found using the iterative algorithm described above (see Table 2.1). For low bit rates, the variance decision levels are reformulated as follows (see e.g., Hjørungnes and Lervik, 1997):

$$\tilde{\sigma}^2_{d,c} = \tilde{\sigma}^2_{d,c,opt} + d_c \times \Delta^2, \quad c = 1, 2, \ldots, C-1, \tag{2.13}$$

where d_c for $c = 1, 2, \ldots, C-1$ are proportionality constants and Δ is the step-size of the uniform threshold scalar quantizer. The employed constants are also provided in Table 2.1.

As an example, the histogram of the subband samples which belong to class 2 (i.e., $c = 2$) at high bit rates is displayed in Figure 2.10. This histogram, as well as the histograms of the subband samples which belong to the other classes, show close resemblance to a Gaussian pdf. Several qualitative experiments indicate that the Gaussian mixture distribution model seems to be appropriate for COGs after subband decomposition. When the block class entries are arranged in a matrix according to the location of the blocks in the subbands, the so-called block classification table is obtained. Figure 2.11 presents the block classification table for the common offset gather after subband decomposition shown in Figure 2.9.

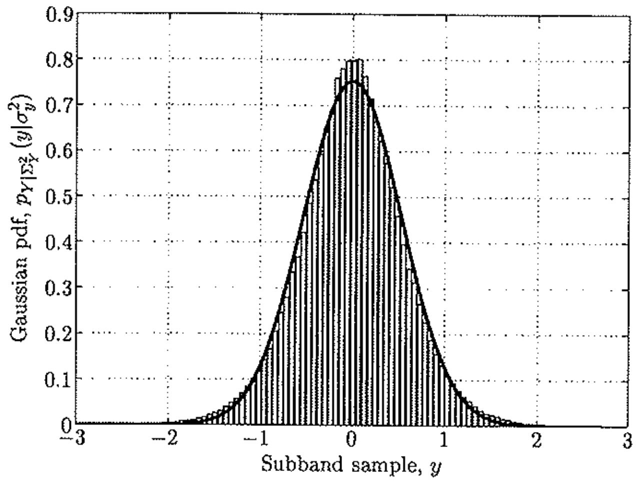


Figure 2.10: Normalized histogram of subband samples which belong to class 2 at high bit rates, that is $\hat{\sigma}_y^2 \in [\tilde{\sigma}_{d,2,opt}^2, \tilde{\sigma}_{d,3,opt}^2)$, and the associated Gaussian pdf (solid line), i.e., $\sim \mathcal{N}(0, \tilde{\sigma}_{r,2,opt}^2)$.

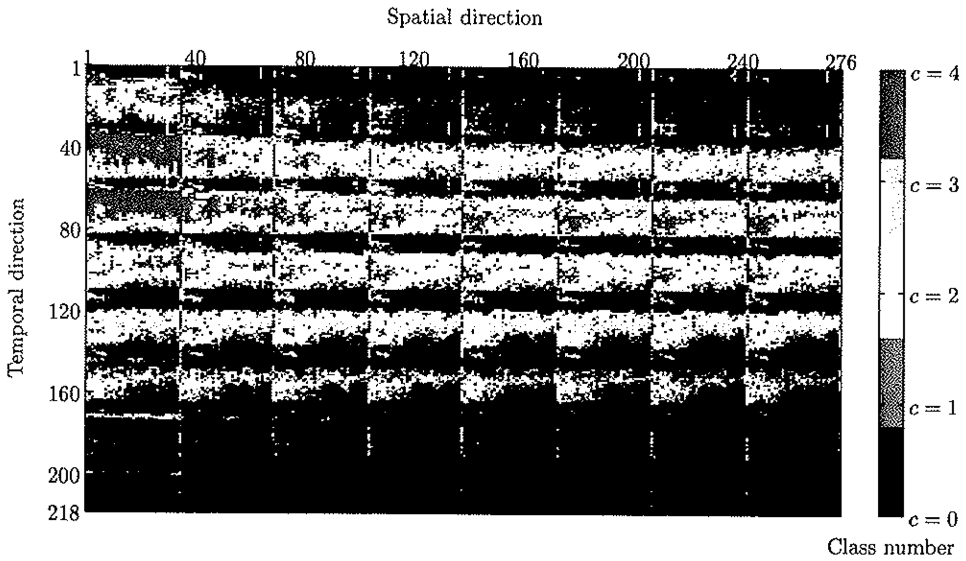


Figure 2.11: Block classification table. The step-size is $\Delta = 0.50$, the block size $\tilde{B} \times \tilde{B} = 4 \times 4$, and the number of classes $C = 5$. The size of the block classification table is (872/4) × (1104/4) = 218 × 276 block class entries (see Figure 2.9).

2.5.3 Structure of the subband coding method

A complete block diagram of our SBC method is displayed in Figures 2.12 and 2.13. The combination of all techniques used in the SBC method represents a unique method. At the encoder, we first sort the common shot gathers to common offset gathers. After RLC of the TWT delay of the water layer and amplitude balancing by an AGC image, the COG is decomposed into $M \times M = 8 \times 8$ subbands by a separable analysis filter bank (Aase, 1993), denoted FB. The AGC mean of absolute values of $B \times B = 32 \times 32$ input samples are coded using a 1-D fixed-rate open-loop DPCM and sent as side information. The 64 subbands are segmented into square blocks of $\tilde{B} \times \tilde{B} = 4 \times 4$ samples from the same subband, and the variance of each block is estimated. Each block is then classified into one out of $C = 5$ classes according to its variance estimate (Lervik, 1996). All subband samples are quantized in one common mid-tread uniform threshold scalar quantizer, denoted SQ.

The presumably uncorrelated subband samples in class number c are coded with a *non-conditional semi-adaptive* multi-alphabet arithmetic encoder (Witten et al., 1987), AC_c, $c = 0, 1, \ldots, 4$, and sent as *main information.* The class histograms are computed for each class and used as probability models for the semi-adaptive arithmetic coders. Hence, the non-zero probabilities must also be sent as side information. Typically, for the four lowest variance classes, approximately 5 % of the possible quantized subband indices have non-zero probability of occurrence. For the highest class, i.e., $c = 4$, typically about 20 % of the quantized subband indices have non-zero probability of occurrence.

An alternative strategy would be to employ fixed (that is non-adaptive) or adaptive arithmetic coders. With these two alternative coding strategies, the non-zero probabilities need not to be sent as side information. Nevertheless, experiments indicate that the use of semi-adaptive arithmetic coders give the best compression results. A possible explanation for the superiority of semi-adaptive arithmetic coders is, for instance, that the probability models for the fixed and adaptive arithmetic coders are not perfectly matched and tuned, respectively.

The block classification table, on the other hand, is coded using a *conditional adaptive* multi-alphabet arithmetic encoder (Nelson and Gailly, 1996), ${}^{\mathcal{C}}\mathrm{AC}$, and sent as side information. Apparently, there is a dependency between block classification table entries from different subbands representing the same area of the original COG (see Figures 2.9 and 2.11). Experiments manifest that using a conditional adaptive arithmetic coder of the block classification table is more efficient than using a straightforward first-order semi-adaptive arithmetic

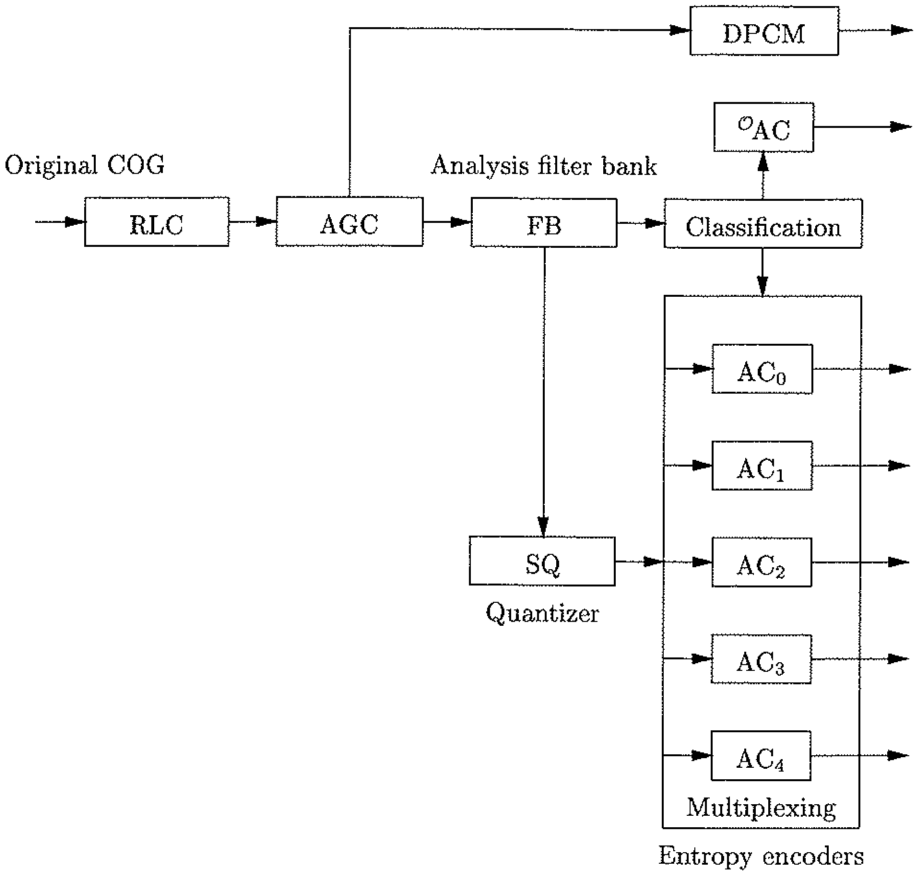


Figure 2.12: The encoder of the utilized SBC method with $C = 5$ classes.

coder. Specifically, we employ a *third-order* (i.e., $\mathcal{O} = 3$) conditional adaptive arithmetic coder of the classification table entries.

At the decoder, the block classification table is decoded by a conditional adaptive multi-alphabet arithmetic decoder, ${}^{\mathcal{O}}\mathrm{AC}^{-1}$. The reconstructed block classification table is used to select the proper non-conditional semi-adaptive multi-alphabet arithmetic decoder, AC_c^{-1}, $c = 0, 1, \ldots, 4$, to decode the quantizer indices for each block. Subsequently, the subband samples are reconstructed in one common inverse mid-tread uniform threshold scalar quantizer, SQ^{-1}. The reconstructed COG is obtained from the $M \times M = 8 \times 8$ subbands by the separable synthesis filter bank, denoted FB^{-1}, followed by inverse AGC, AGC^{-1}. The AGC mean of absolute values are achieved by a 1-D inverse fixed-rate open-loop DPCM, DPCM^{-1}, and the complete AGC image is retrieved

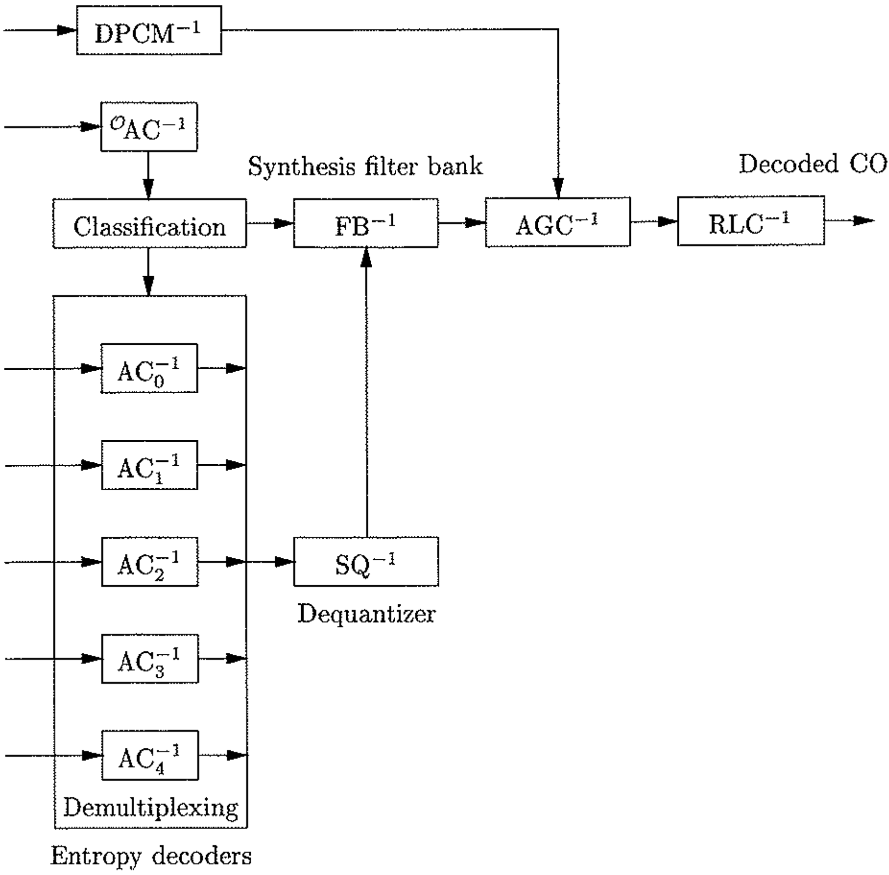


Figure 2.13: The decoder of the utilized SBC method with $C = 5$ classes.

by piecewise-linear interpolation applied separately in the temporal and spatial directions. The TWT delay of zero amplitude values is put back on the COG at the end. Finally, we resort from COGs to CSGs.

2.6 Compression results

In this section, we test our SBC algorithm on the seismic data set displayed in Figure 2.1. Specifically, the data set is a COG at medium offset with 1101 number of shots and 1001 number of samples per trace. The sampling interval in time and space is 4.0 ms and 12.5 m, respectively. Using this COG example, we compare our SBC algorithm (named system **I**) with three other seismic data compression techniques: two non-commercial algorithms [system **II** (GenLOT;

Duval et al., 1999b) and system **III** (Seismic Unix; Chen, 1995)] and one commercial algorithm [system **IV** (ProMAX; Donoho et al., 1995)].

System **II** is based on parallel uniform 8-channel 32-tap GenLOT signal decomposition with uniform SQ and so-called embedded zero-tree coding (EZC; Shapiro, 1993) using non-conditional adaptive AC. The GenLOTs are adapted to the correlation of COGs using the AR processes suggested in this work (see Duval and Røsten, 2000). In EZC, the "most important" quantizer indices are sent from the encoder to the decoder first. An interesting feature of EZC is progressive compression. Embedded means that the encoder can stop encoding at any desired (integer) target bit rate with approximately optimal rate-distortion performance. Similarly, the decoder can stop decoding at any desired target bit rate. This property is for instance appealing for quality control of seismic data. The comparison with system **II** is not quite "fair" since it is implemented to compress seismic data having 32 bits unsigned-long-integer number representation. Thus, an extra quantization error is introduced in the conversion from the floating-point to the fixed-point number representation at the encoder, and vice versa at the decoder. Anyhow, this extra quantization error is small and a fairly good comparison can be made.

System **III** is the "wptcomp"/"wptuncomp" software included in the Seismic Unix package (see e.g., Stockwell, 1997) which is based on tree-structured full-split (or uniform) orthogonal wavelet signal decomposition, with uniform SQ and HC. System **IV** is installed as a so-called third-party software on ProMAX and based on tree-structured dyadic biorthogonal wavelet signal decomposition, with uniform *adaptive* SQ followed by RLC and HC. We demonstrate the performance of our SBC algorithm compared to the three other seismic data compression techniques with both quantitative and qualitative results.

The same preprocessing is utilized by all four compression systems, i.e., we compress the COG after RLC and AGC (see Figures 2.3 and 2.12). Thus, the necessary amount of side information for this preprocessing is equal for all four compression schemes. Immediately after compression and decompression [including AGC^{-1} and RLC^{-1} (see Figure 2.13)], we calculate the full signal-to-quantization-noise ratio (SQNR; Vermeer, 1999) between the original and the decoded COG, and use the SQNR as a quantitative measure of compression quality. We use the term SQNR to distinguish it from the more conventional expression signal-to-noise ratio (SNR). The SNR is commonly used for the purpose of quality measure of seismic data in the presence of ambient noise (see e.g., Junger, 1964).

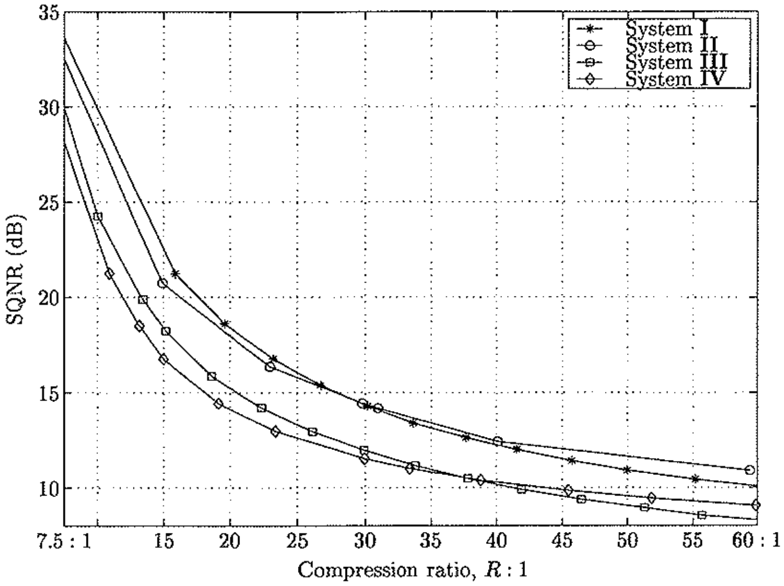


Figure 2.14: SQNR as a function compression ratio for the four systems.

The SQNR - in dB - is defined as:

$$\mathrm{SQNR} = 20 \times \log_{10}\left(\frac{\mathrm{rms}}{\mathrm{rmse}}\right), \tag{2.14}$$

where rms is the root-mean-square value of the original seismic data and rmse is the root-mean-square error between the original and the decompressed seismic data. The SQNR is used as a quality measure although the SQNR is highly dependent on the bandwidth characteristics and the amplitude distribution (or balance) of the data set. For example, a narrow bandwidth compressed-decompressed data set normally has higher SQNR than a broad bandwidth dataset. A well balanced compressed-decompressed data set tends to have lower SQNR than a poorly balanced data set even though the actual quality of well balanced data sets is better (Reiter, 1996). Nevertheless, the SQNR provides a good relative indication of the preferred grading between different compression algorithms.

Figure 2.14 shows the resulting SQNR as a function of compression ratio (CR), ranging from 7.5:1 to 60:1, for the four different compression algorithms. The CR is simply defined as the size of the trace data samples before compression, divided by the size of the trace data samples after compression. Thus, a CR of $R:1$ corresponds to $32/R$ bits per sample. System **I** and system **II** have more or less identical performance, but they significantly outperform the

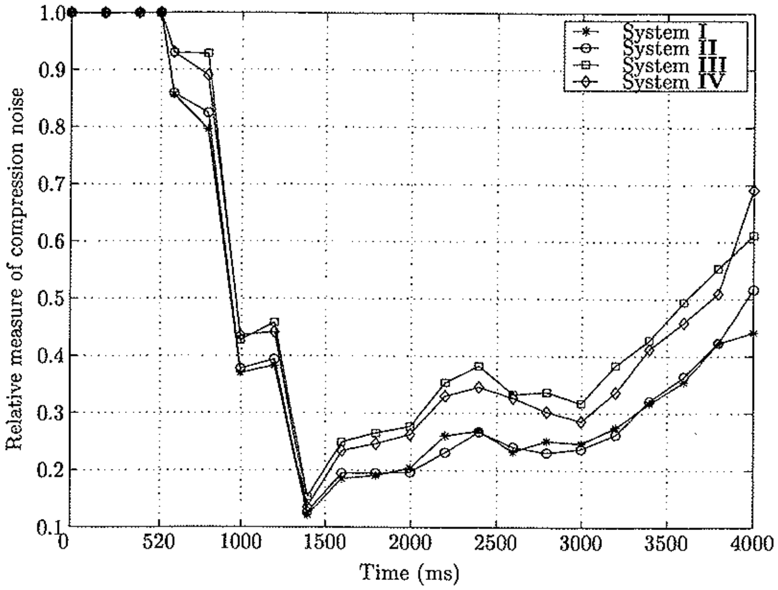


Figure 2.15: Measure of compression noise versus time for the four systems.

two other compression methods at all compression ratios. For example, at an SQNR equal to 15 dB, system **I** and system **II** perform at least 30 % better than the two other compression algorithms in terms of CR.

A relative measure of compression noise as a function of time is computed by dividing the rmse between the original and the decompressed COGs – rowwisely – by the rms value of the original COG (see Figure 2.15). The CR is approximately 30:1 for all four compression schemes. System **I** and system **II** have the lowest relative amounts of compression noise at all times. The amplitudes just prior to the first break (or the TWT delay of the water layer) provides an estimate of ambient noise in the data set (Donoho et al., 1999a,b, 2000). As can be seen, in the target zone below the TWT delay of the water layer, i.e., from $(132 - 1) \times 4 = 524$ ms, the relative measure of compression noise versus time is always lower than the relative measure of compression noise (or ambient noise) in the shallow zone (being equal to 1.0). This is reasonable since the amplitudes in this shallow zone are set to zeros during the compression.

The energy of the compression noise can be larger than the energy of the ambient noise (Donoho et al., 1999b), at least for high compression ratios. Figure 2.16 shows the histogram of the estimated ambient noise (i.e., pre-first-break arrivals) and the histograms of the compression noise for the four

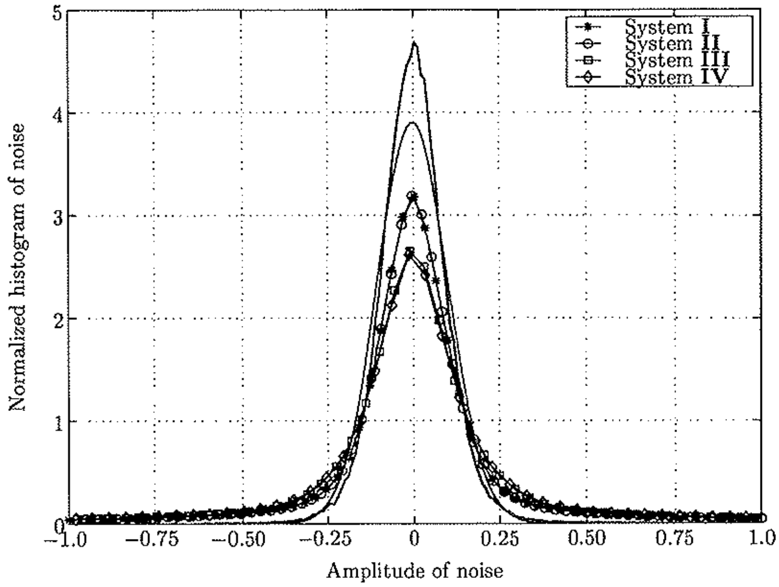


Figure 2.16: Normalized histograms of the compression noise for the four systems, histogram of the estimated ambient noise (blue line), and the associated Gaussian pdf (red line), i.e., $\sim \mathcal{N}(0, 0.01)$.

compression systems at a CR of 30:1. A common assumption is that the ambient noise can be modeled as additive white Gaussian noise. The pre-first-break arrivals closely exhibit a Gaussian pdf for the example (compare the blue line and the red line in Figure 2.16). Clearly, at a CR of 30:1, the compression noise is somewhat larger in average than the ambient noise. In addition, the histograms of the compression noise have long tails, indicating non-Gaussian pdfs. Figure 2.17 plots the rms levels of the compression noise as a function of CR, ranging from 7.5:1 to 60:1, for the four compression systems. This diagnostic metric has been suggested by Donoho et al. (1999b). As can be seen, the rms levels of the compression noise are larger than the rms level of the ambient noise at all compression ratios from 10:1 to 60:1. In the range from 7.5:1 (for systems **III** and **IV**) to 10:1 (for systems **I** and **II**), however, the rms levels of the compression noise strongly approach (or become lower than) the rms level of the ambient noise. At the same time, the histograms of the compression noise approach Gaussian pdfs for all four systems.

Closeup views in the time window from 1252 to 2752 ms of the 30:1 decompressed COGs using system **I**, **II**, **III**, and **IV** are shown in panel (b) of Figures 2.18, 2.19, 2.20, and 2.21, respectively. The original section is for convenience given in panel (a) of all four figures. Very detailed and good recon-

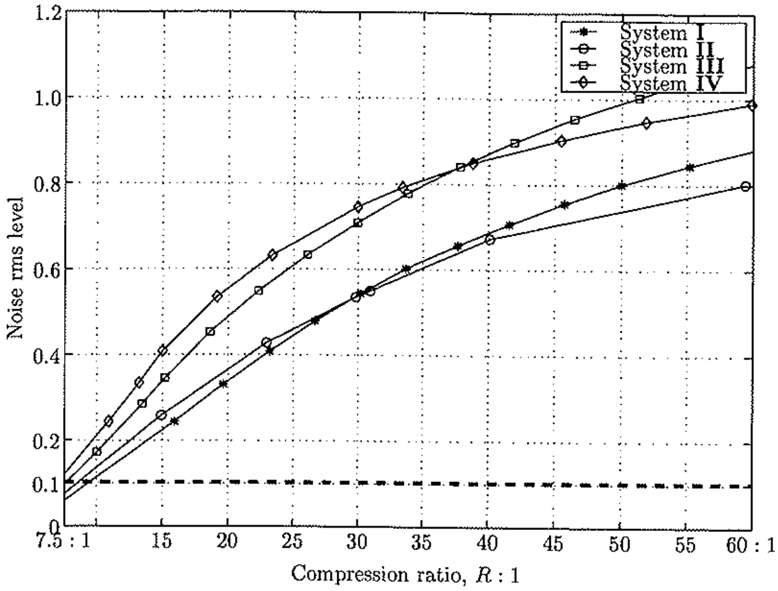


Figure 2.17: Compression-noise rms levels for the four systems as a function of compression ratio, and the rms level of the estimated ambient noise (dashed line).

structions are provided; the original and the reconstructed sections look almost identical. In fact, the decoded COGs appear more distinct compared to the original common offset gather. The difference sections, gained by a factor of 5 (see panel (c) of Figures 2.18 to 2.21), exhibit mostly the presence of "uncorrelated" compression noise, even though the examined piece is taken near the water bottom, where the largest differences are observed. However, systems **I** and **II** have lower amounts of, and probably less correlated, compression error compared to systems **III** and **IV**.

2.7 Discussion

We have used a 2-D AGC to decrease the non-stationary behavior of seismic data prior to compression. Alternatively, we can gain the traces individually as a function time t by applying a time-power constant $\delta > 1.0$, i.e., t^δ. Appropriate values of δ can be estimated from trace to trace, but these values must be compressed and sent as side information. The main drawback with this approach, however, is that the head waves, that arrive after the first break, can have very high amplitudes. A t^δ-strategy usually blows these refractions up out of all proportions (at least for large offsets) since head waves do not decay according to such a overly simplistic gain function. The application of an outer mute prior to compression to remove head waves is thus desired.

The complexity of the proposed SBC scheme is of the same order of magnitude as the JPEG standard (see e.g., Nelson and Gailly, 1996) and the dyadic biorthogonal wavelet-based seismic data compression algorithm of Donoho et al. (1995). The performance of the JPEG standard is, however, poor for seismic data due to blocking artifacts. The complexity increase due to the classification procedure of our SBC method is somewhat compensated for by avoiding RLC as used by Donoho et al. (1995). The complexity of the algorithms presented by Chen (1995) and Duval et al. (1999b) are slightly lesser and higher than our SBC algorithm, respectively.

Tests indicate that optimization of the filter banks and optimization of the variable-rate arithmetic coders are more or less of equal importance. Using the filter banks optimized for still images (Aase, 1993), that is AR(1) model with $\rho_1 = 0.95$ employing the same filter bank properties as for seismic data and using one single arithmetic coder, our SBC method still has better performance than the two wavelet-based seismic data compression algorithms of Chen and Donoho et al. Tests indicate (see Duval and Røsten, 2000) that our non-unitary filter banks perform slightly better than the orthogonal GenLOTs used by Duval et al..

2.8 Conclusions

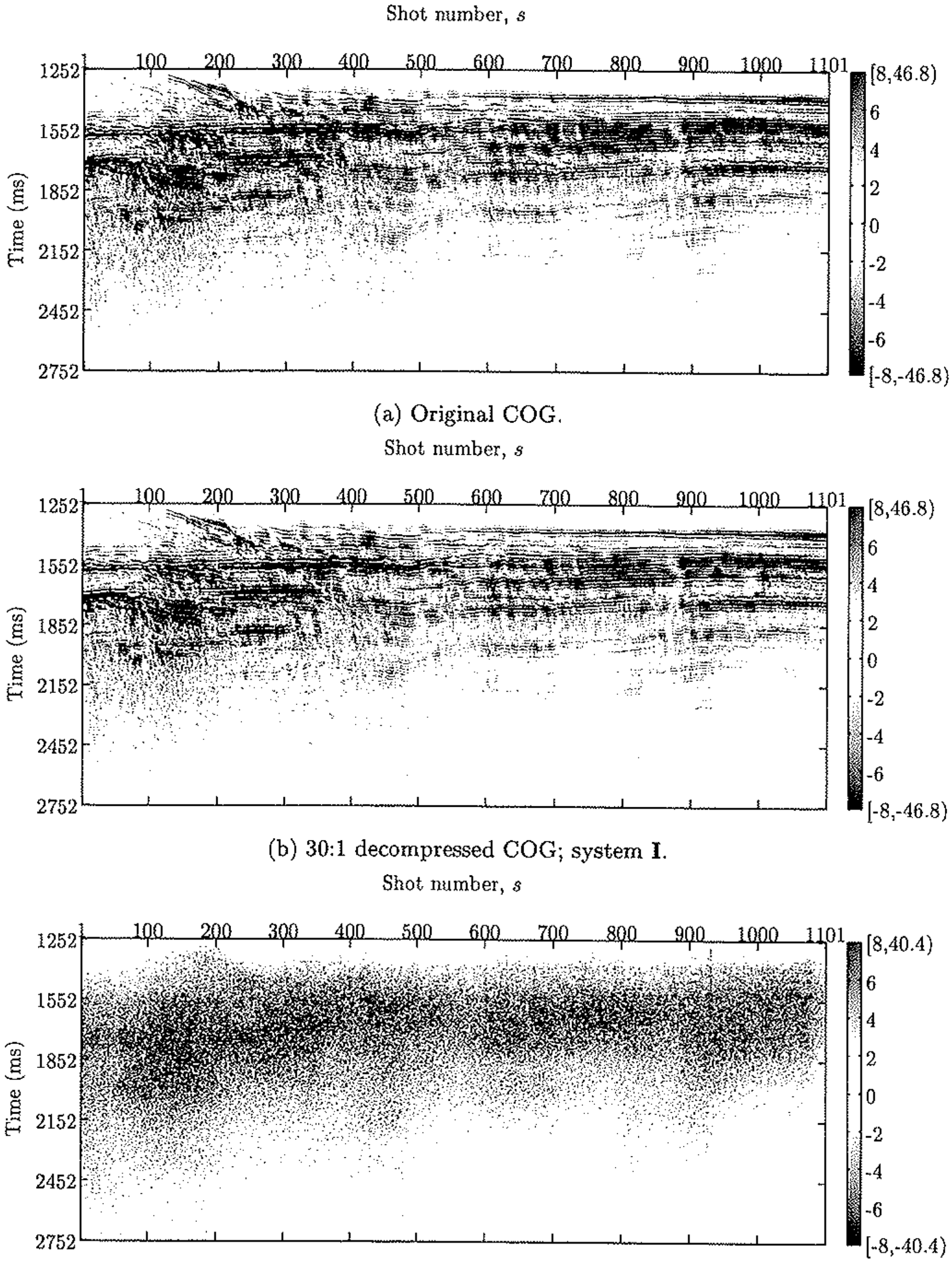
A 2-D lossy subband coding scheme for seismic data has been presented. We have focused on the compression of common offset gathers since they have higher coherency, and therefore are more suited for compression, than for example common shot gathers. The introduced compression noise is more or less equal to, and has similar characteristics as, ambient noise components at moderate compression ratios (around 10:1) for the COG example.

From quantitative and qualitative compression results we find that the proposed subband coding algorithm has excellent performance compared to three other seismic data compression algorithms. Nevertheless, no definitive preference between the four different seismic data compression algorithms can be made due to the limited number of examples.

Due to its good compression results and low complexity, the proposed subband coding method is a very promising approach which can be easily extended to high-dimensional compression of prestack and poststack seismic data.

2.9 Acknowledgments

Tage Røsten thanks Den norske stats oljeselskap a.s (Statoil) and Petroleum Geo-Services (PGS) for financial support. We appreciate helpful discussions with Laurent Duval and acknowledge him for providing the GenLOT seismic data compression algorithm. The authors thank Dr. Gijs Vermeer and Associate Editor Panos G. Kelamis for their very thorough and helpful review of this manuscript.

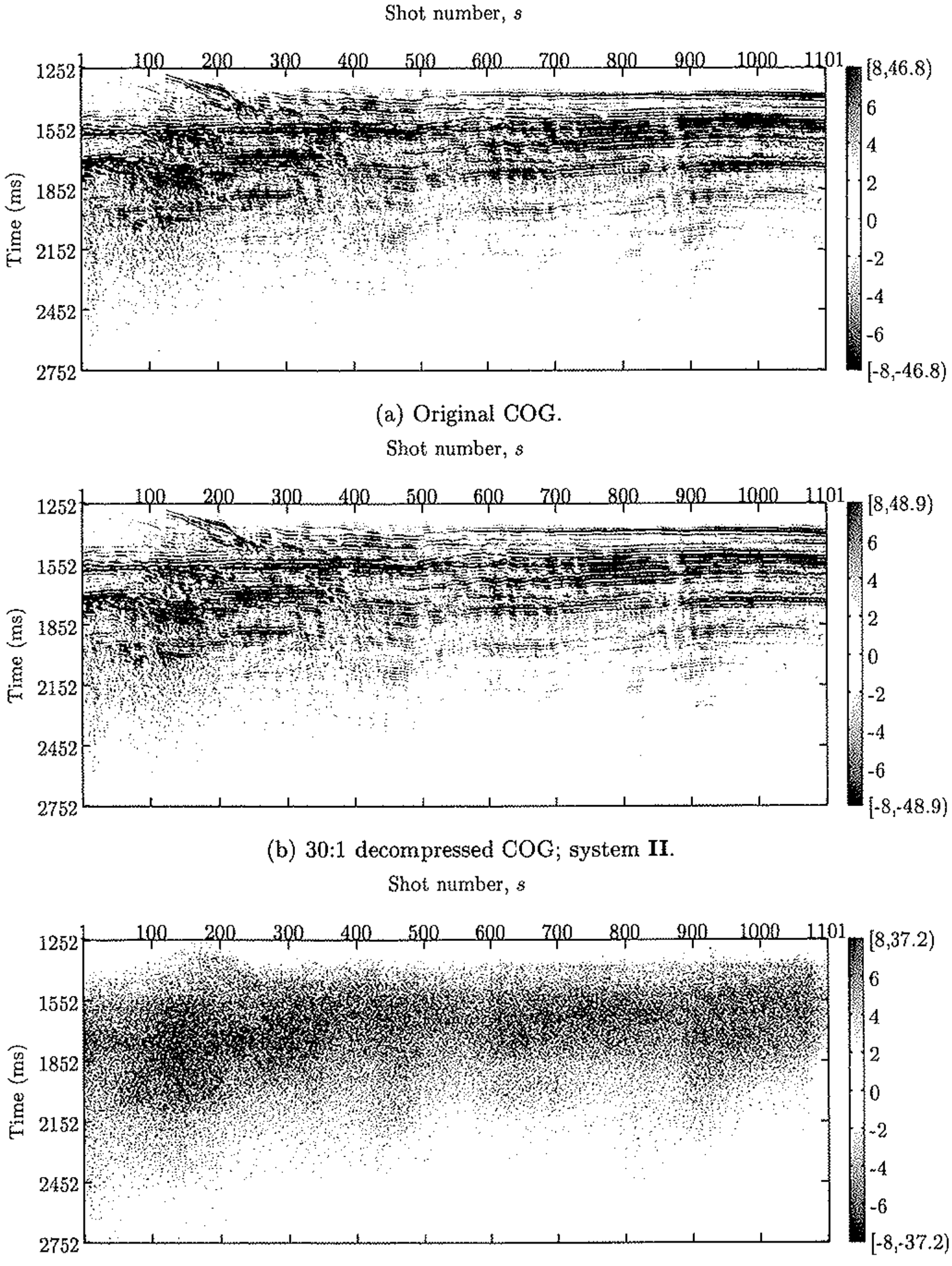


(a) Original COG.

(b) 30:1 decompressed COG; system **I**.

(c) Difference (×5) between (a) and (b).

Figure 2.18: Closeup views of reconstructed and difference COGs; system **I**.

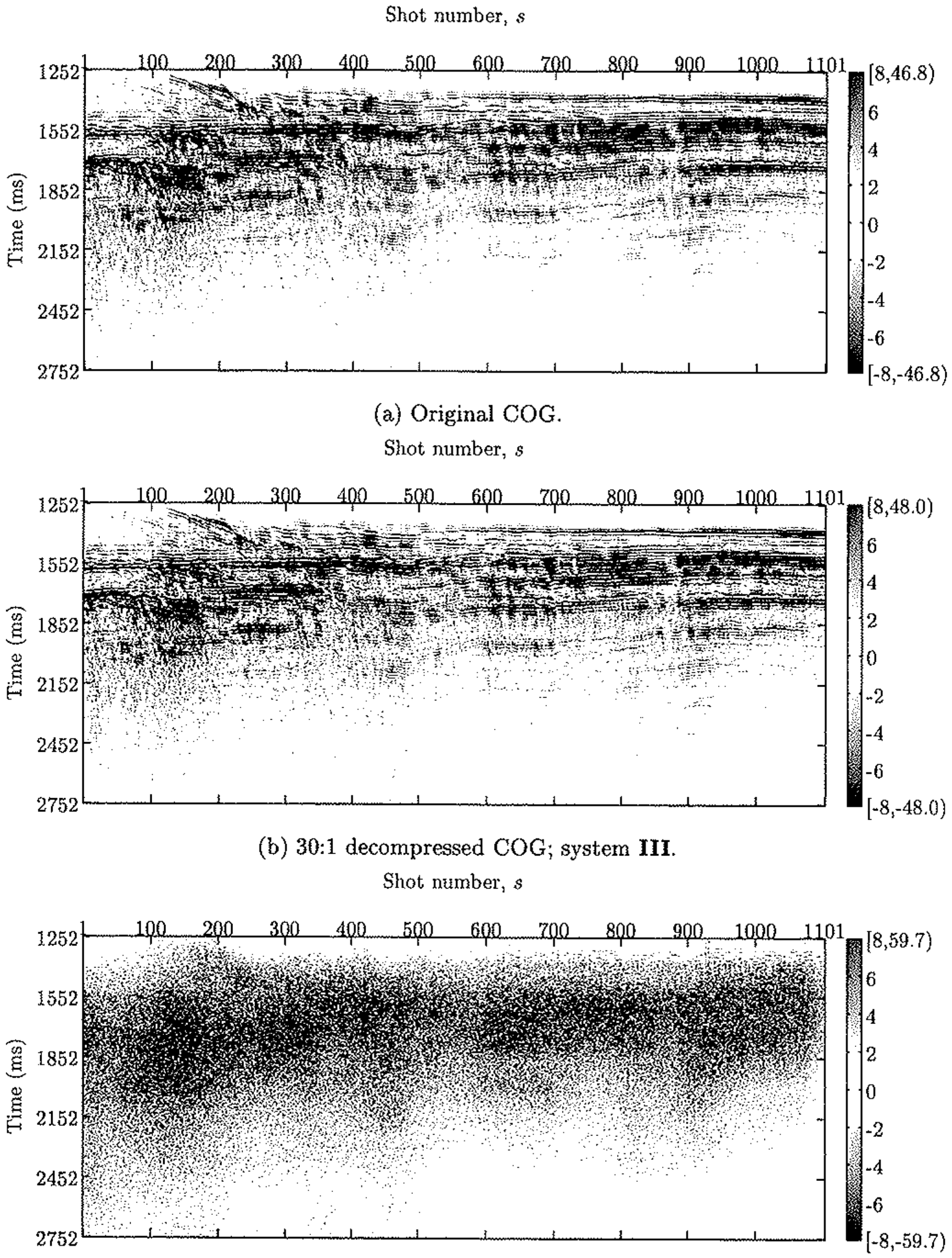


(a) Original COG.

(b) 30:1 decompressed COG; system **II**.

(c) Difference (×5) between (a) and (b).

Figure 2.19: Closeup views of reconstructed and difference COGs; system **II**.

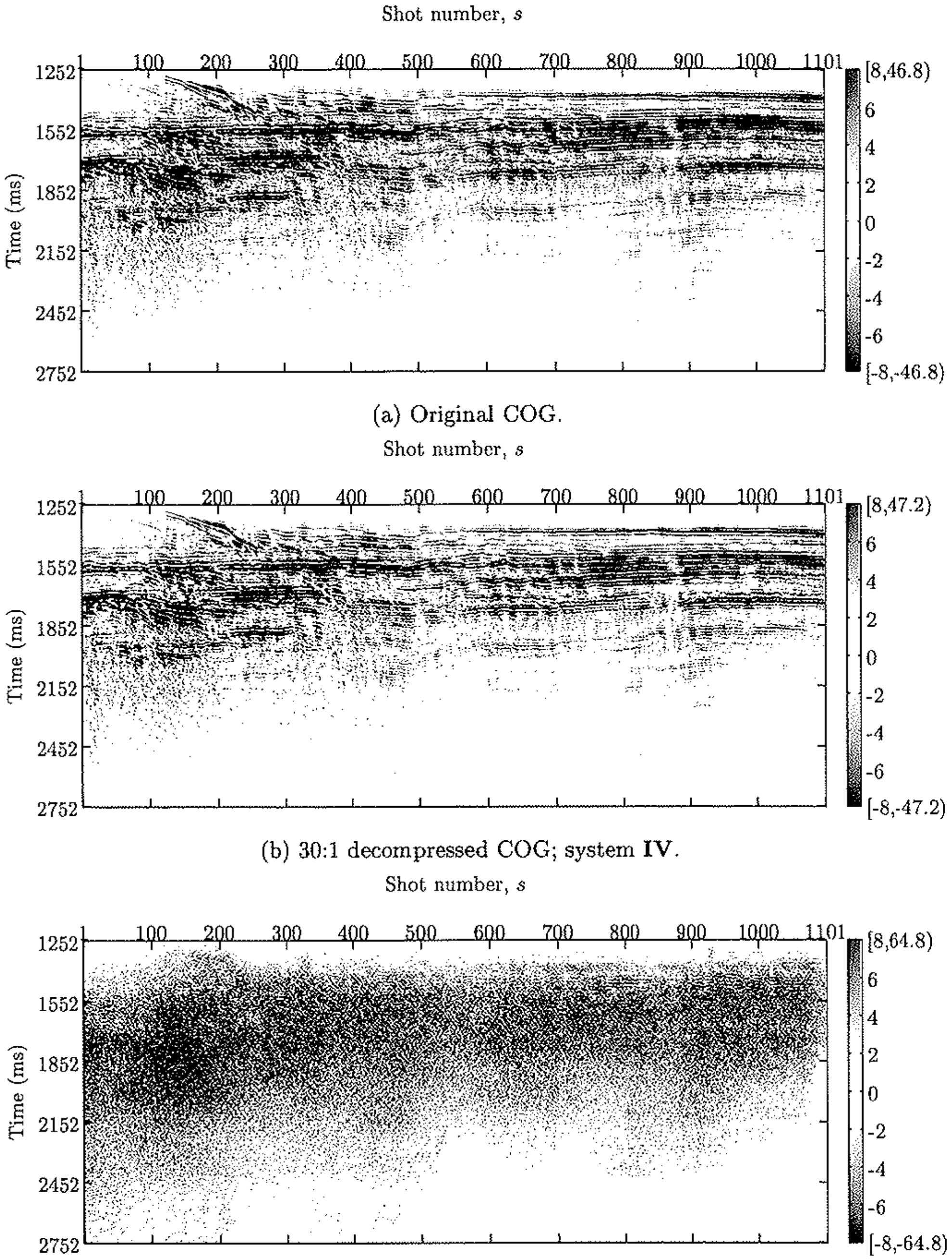


(a) Original COG.

(b) 30:1 decompressed COG; system **III**.

(c) Difference (×5) between (a) and (b).

Figure 2.20: Closeup views of reconstructed and difference COGs; system **III**.



(a) Original COG.

(b) 30:1 decompressed COG; system **IV**.

(c) Difference (×5) between (a) and (b).

Figure 2.21: Closeup views of reconstructed and difference COGs; system **IV**.

2.A Autocorrelation functions of AR processes

The ACFs of first-order and second-order real-valued Gauss-Markov processes are given in Jayant and Noll (1984). The 1-D input sample, $x(n)$, is assumed to be characterized by a zero-mean unit-variance real stochastic variable, $x \in X$.

2.A.1 AR(1) process

The ACF of an AR(1) process is

$$r_{xx}(n) = \rho_1^{|n|}, \tag{2.A-1}$$

where ρ_1 denotes the normalized correlation coefficient at lag one.

2.A.2 AR(2) process

The ACF of an AR(2) process is recursively given by

$$r_{xx}(n) = \begin{cases} 1.0, & n = 0 \\ \phi r_{xx}(n-1) + \psi r_{xx}(n-2), & n \geq 1 \\ r_{xx}(-n), & n < 0. \end{cases} \tag{2.A-2}$$

The constants ϕ and ψ in equation (2.A-2) are given by

$$\phi = \frac{\rho_1(1-\rho_2)}{1-\rho_1^2} \quad \text{and} \quad \psi = \frac{\rho_2 - \rho_1^2}{1-\rho_1^2}, \tag{2.A-3}$$

where ρ_2 denotes the normalized correlation coefficient at lag two.

2.B Filter coefficients

The filter coefficients of the 8-channel 32-tap uniform filter banks used in this work are given in Tables 2.B-2-2.B-5. The analysis $h_m(l)$ and the synthesis $g_m(l)$ filter coefficients for $m = 0, 1, \ldots, 7$ and $l = 0, 1, \ldots, 31$ are first given in the temporal direction, and then in the spatial direction. Remark, only the first half of the filter coefficients are tabulated. The second half of the filter coefficients are given by even or odd symmetry since the FIR filter banks have linear phase: even symmetry for m even and odd symmetry for m odd.

2.B.1 Error function

The filter banks are optimized using suitable error terms to approach PR, removal of blocking effects, and high coding gain. The requirement of linear phase filters is an additional constraint. A suitable error function for our purpose is (Ramstad et al., 1995)

$$\varepsilon = w_P \varepsilon_P + w_B \varepsilon_B + w_G \varepsilon_G, \tag{2.B-1}$$

where the error terms are included to account for PR (ε_P), blocking effect (ε_B), and coding gain (ε_G). To find the expressions and the associated derivatives for the terms in the error function is a comprehensive task, and full details can be found in Aase (1993). For example, the error term corresponding to coding gain is given by (see equation 2.8):

$$\varepsilon_G = \left[\prod_{m=0}^{M-1} A_m \right]^{1/M} . \tag{2.B-2}$$

As shown in equation 2.B-1, each error term is multiplied by a proper weight factor. A suitable choice of weight factors for our purpose is $w_P = 100.0$, $w_B = 1000.0$, and $w_G = 1.0$. These respective weight factors were found by a systematic trial and error procedure.

The error function of equation 2.B-1 is minimized through a gradient search method, and hence we can not claim to find definite optimal filter banks. For faster convergence and for reducing the risk of falling into a local optimum, a proper filter bank system is used as an initial filter bank in the optimization procedure. We have used the DCT with $M = L = 8$ as a starting point for the iterations. Since the optimized filter banks should have $M = 8$, but $L = 32$, the initial filter coefficients outside the 8-tap center of the DCT filter coefficients are set to zero.

2.B.2 Coding gain results

The *maximum theoretical* coding gain, $\tilde{G}_{\text{SBC}}$, can be found assuming a uniform filter bank having infinite number of channels and infinite number of filter taps in all channels. The $\tilde{G}_{\text{SBC}}$, as well as the G_{SBC} (see equations 2.8 and 2.10), depends on the ACF of the input signal (Jayant and Noll, 1984). The maximum theoretical coding gain for a zero-mean unit-variance AR(1) process is given by

$$\tilde{G}_{\text{SBC}}^{\text{AR(1)}} = \left(1 - \rho_1^2\right)^{-1}. \tag{2.B-3}$$

Similarly, a zero-mean unit-variance AR(2) process has maximum theoretical coding gain equal to

$$\tilde{G}_{\text{SBC}}^{\text{AR(2)}} = \left(1 - \rho_1^2 - \frac{(\rho_1^2 - \rho_2)^2}{1 - \rho_1^2}\right)^{-1}. \tag{2.B-4}$$

The resulting practical and maximum theoretical coding gains – in dB – for the optimized filter banks are tabulated in Table 2.B-1. For comparison, the resulting practical coding gains for the 8-channel DCT, assuming similar AR processes as for the optimized filter banks, are also tabulated. Significantly improvements for the optimized filter banks compared to the DCT are achieved. The coding gain decreases for decreasing L. On the other side, the DCT is close to the optimal transform [i.e., the discrete Karhunen-Loève transform (DKLT)] for an AR(1) process with ρ_1 approaching unity (Ramstad et al., 1995).

Table 2.B-1: Practical and maximum theoretical coding gains. The G_{SBC} for the 8-channel DCT are for comparison enclosed within parentheses.

	Coding gains (dB)	
	Practical, G_{SBC}	Maximum, $\tilde{G}_{\text{SBC}}$
Filter		
8-32-I_062_010	3.07 (2.45)	3.15
8-32-I_070	2.87 (2.50)	2.92

Table 2.B-2: Analysis filter coefficients in the temporal dir.; 8-32-I_062_010.

	Analysis filter coefficients, $h_m(l)$							
	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
l								
0	0.006096	-0.015941	-0.023007	-0.008099	0.011024	-0.003653	-0.007104	-0.007900
1	-0.009134	0.000476	0.003330	0.005960	-0.021504	-0.004128	0.014532	0.016889
2	0.000601	0.017705	0.031131	0.007647	0.011982	0.013646	-0.008893	-0.010511
3	0.000165	-0.009817	-0.005237	-0.000981	0.010104	0.002960	-0.010908	-0.012643
4	-0.015298	-0.067306	-0.080253	-0.012672	-0.027305	-0.018986	0.029557	0.032331
5	-0.008286	-0.065844	-0.064885	-0.007899	0.030307	0.003832	-0.029121	-0.031880
6	-0.025789	-0.007358	0.015607	0.019643	0.000909	0.019207	0.005515	0.001630
7	-0.052737	0.001028	0.024532	0.033409	-0.065426	0.004166	0.035082	0.051029
8	-0.037663	-0.103766	-0.038451	0.118896	0.078424	-0.112463	-0.078470	-0.071138
9	0.033872	-0.139232	-0.178099	-0.044076	0.111467	0.231373	0.075029	0.031106
10	0.100825	-0.036716	-0.151063	-0.284211	-0.244996	-0.175210	0.020769	0.048119
11	0.152293	0.153597	0.124966	-0.059419	-0.039820	-0.096330	-0.189972	-0.138138
12	0.213064	0.332400	0.393746	0.364026	0.380032	0.354140	0.355229	0.228646
13	0.284517	0.421830	0.351936	0.166808	-0.124918	-0.306582	-0.426396	-0.312698
14	0.369063	0.360976	-0.018324	-0.397619	-0.416913	-0.059937	0.351563	0.379348
15	0.447614	0.145986	-0.387658	-0.322158	0.306708	0.419064	-0.136123	-0.418129

Table 2.B-3: Synthesis filter coefficients in the temporal dir.; 8-32-I_062_010.

	Synthesis filter coefficients, $g_m(l)$							
	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7
l								
0	-0.000160	0.003387	-0.004043	-0.003410	-0.000019	0.008453	-0.001746	0.000671
1	0.001048	-0.007547	0.008759	-0.012156	-0.014173	0.000314	0.016516	-0.022710
2	0.003908	-0.012832	0.013918	0.002350	0.009395	-0.017636	-0.014318	0.019351
3	0.006465	0.012661	-0.015430	0.018354	0.005434	0.005030	-0.008677	0.013456
4	-0.000750	0.040950	-0.045001	0.019606	-0.018034	0.032693	0.030401	-0.040645
5	-0.021309	0.029873	-0.024888	0.002297	0.024577	-0.000579	-0.030727	0.041260
6	-0.048609	0.001984	0.015180	-0.049628	0.007158	-0.020685	0.011331	-0.016950
7	-0.065727	0.024934	-0.002096	-0.093352	-0.018279	0.026760	0.033094	-0.055269
8	-0.043934	0.109218	-0.099950	-0.075215	0.101385	0.063607	-0.078469	0.099161
9	0.013576	0.147900	-0.188223	0.137132	0.042723	-0.221666	0.097492	-0.038866
10	0.078129	0.062944	-0.102608	0.277469	-0.244312	0.202597	-0.005522	-0.047053
11	0.136482	-0.124655	0.169837	-0.009494	0.029259	0.071738	-0.169810	0.135495
12	0.200938	-0.322868	0.396283	-0.354029	0.367703	-0.347160	0.346933	-0.224089
13	0.282134	-0.427550	0.321291	-0.109313	-0.168587	0.321836	-0.426406	0.308310
14	0.376453	-0.367232	-0.043690	0.400296	-0.404493	0.041150	0.360319	-0.373183
15	0.451957	-0.145624	-0.386818	0.298105	0.313810	-0.414986	-0.140746	0.419260

Table 2.B-4: Analysis filter coefficients in the spatial dir.; 8-32-I_070.

	Analysis filter coefficients, $h_m(l)$							
	h_0	h_1	h_2	h_3	h_4	h_5	h_6	h_7
l								
0	0.028521	0.016711	-0.015109	0.005772	0.009862	0.010154	0.000174	-0.001362
1	0.029985	0.008172	0.016105	0.016532	-0.000500	-0.006698	-0.000360	0.002941
2	0.014923	-0.003037	0.029721	0.013845	-0.011234	-0.018105	-0.001968	0.003341
3	-0.018072	-0.010483	0.003989	-0.008920	-0.012195	-0.010779	-0.002152	-0.002779
4	-0.051874	-0.018327	-0.024486	-0.019075	0.036217	0.040724	0.016813	0.014945
5	-0.063421	-0.025978	-0.021024	-0.026191	-0.017950	-0.004222	-0.030814	-0.037034
6	-0.047956	-0.025749	0.036362	-0.018587	-0.025978	-0.028487	0.033931	0.035760
7	-0.028350	-0.035764	0.071440	0.030709	0.011984	-0.008299	-0.013524	0.007132
8	-0.065276	-0.121619	-0.039981	0.147132	0.030011	-0.010965	-0.037661	-0.055654
9	-0.029414	-0.158878	-0.201710	-0.079363	0.117220	0.138888	0.060909	0.050566
10	0.041005	-0.086445	-0.156086	-0.282150	-0.224415	-0.138755	0.013513	-0.000556
11	0.139722	0.094500	0.111377	0.011158	-0.085653	-0.090311	-0.175982	-0.098496
12	0.243833	0.308314	0.361981	0.371695	0.385712	0.368322	0.350400	0.213524
13	0.326550	0.433381	0.315170	0.137313	-0.097106	-0.344411	-0.433517	-0.319338
14	0.380350	0.385782	-0.053417	-0.383178	-0.420351	-0.045989	0.361702	0.392374
15	0.409575	0.161938	-0.433829	-0.302472	0.303973	0.442656	-0.141355	-0.425021

Table 2.B-5: Synthesis filter coefficients in the spatial dir.; 8-32-I_070.

	Synthesis filter coefficients, $g_m(l)$							
	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7
l								
0	0.007463	-0.003982	0.003156	-0.017577	0.009358	-0.003249	-0.000355	-0.003806
1	0.005317	0.008598	0.018240	-0.024589	-0.002268	0.015302	-0.001474	-0.007345
2	-0.005030	0.021921	0.021601	-0.013073	-0.013063	0.024722	-0.002316	-0.005984
3	-0.019517	0.027365	0.002146	0.019659	-0.010059	0.005402	-0.000390	0.003258
4	-0.029959	0.031790	-0.017216	0.036616	0.032097	-0.050018	0.016001	-0.009592
5	-0.031347	0.040529	-0.010212	0.035013	-0.018171	0.004480	-0.029824	0.036694
6	-0.025098	0.049690	0.032128	0.007768	-0.027267	0.040574	0.034469	-0.042506
7	-0.013592	0.065095	0.042921	-0.052878	0.016627	0.010200	-0.019031	-0.004106
8	-0.004361	0.106162	-0.060483	-0.105525	0.054347	-0.020255	-0.038693	0.070838
9	0.035701	0.102718	-0.174097	0.121678	0.088697	-0.144998	0.070920	-0.051181
10	0.098056	0.011205	-0.100492	0.278828	-0.237640	0.150349	0.002003	0.009907
11	0.176120	-0.157290	0.157748	-0.032956	-0.063036	0.066600	-0.166805	0.095769
12	0.254916	-0.335560	0.371595	-0.369249	0.381299	-0.362242	0.347189	-0.210671
13	0.319950	-0.424899	0.292052	-0.115333	-0.113183	0.348068	-0.434200	0.321266
14	0.365755	-0.359145	-0.081206	0.389928	-0.415625	0.035775	0.365051	-0.391057
15	0.392123	-0.143210	-0.443834	0.294352	0.308292	-0.438996	-0.143558	0.423316

Paper 2

Chapter 3

Seismic data compression, Part II: Lossless coding of trace identification headers

3.1 Abstract

Trace headers are an integral part of seismic data. Uncompressed trace headers become a significant portion of the total volume of seismic data in case the associated trace data samples are compressed. We present an efficient lossless compression algorithm applicable for the trace headers of the SEG-Y standard. The proposed technique uses differential coding and run-length coding in conjunction with class-wise entropy coding of the symbols (run) and the counts (length). Four entropy coding methods are compared: non-conditional semi-adaptive Huffman coding, non-conditional adaptive Huffman coding, non-conditional semi-adaptive arithmetic coding, and finally, conditional adaptive arithmetic coding. Typical compression ratios, in the case of trace headers of the common offset gather examples, are about 25:1 for the three non-conditional entropy coding methods and around 250:1 for the conditional entropy coding method. For low and medium orders, the compression ratio increases monotonically as a function of the order of the conditional probability model. The best suited order is 8. After compression, the size of trace headers becomes, independently of the applied entropy coding method, negligible compared to the size of the associated trace data samples.

3.2 Introduction

Acquisition of seismic surveys generates huge amounts of data. Despite the fact that more effective storage and transmission capacities are also introduced, there is an increasing demand for data compression techniques to make the seismic data handling more efficient. In previous work on seismic data compression, emphasis has been on *lossy compression of the trace data samples* (see e.g., Luo and Schuster, 1992; Bosman and Reiter, 1993; Donoho et al., 1995; Røsten et al., 1996, 1997a). Lossless compression of trace data samples can only offer compression ratios in the order of 2:1. Trace headers are an integral part of seismic data, and lossless compression of trace headers can achieve significantly higher compression ratios than 2:1. This is because trace headers contain frequently repeated letters and patterns of letters while trace data samples typically are far less structured (Reiter and Heller, 1994).

A complete seismic data compression system needs to incorporate *lossless compression of the trace headers* preceding the trace data samples. We have to use lossless compression in the case of trace headers since seismic data processing problems else would arise from incorrect reconstructed field geometry. If we do not compress the trace headers, but do compress the trace data samples, the trace headers become a significant portion of the total volume of the data set (Diller et al., 1996). If we for simplicity assume that that trace headers occupy 10 % of an original data set, the uncompressed trace headers start to "dominate" (that is, become the largest part of) the data set when the trace data samples are compressed more than 9:1 (see Figure 3.1). Trace headers are compressed by a factor of at most 10:1 if standard non-matched lossless compression methods are applied (Villasenor et al., 1996). Thus, increasing the compression ratio of trace headers is attractive since compression ratios significantly greater than 90:1 of trace data samples would otherwise provide less marginal coding efficiency.

Recently, we proposed a lossless compression algorithm of trace headers (Røsten et al., 2000a) based on (zero-order) non-conditional semi-adaptive arithmetic coding. However, we have showed that (higher-order) conditional adaptive arithmetic coding significantly increases the compression efficiency of trace headers (Røsten and Ramstad, 1999). In this work, we compare the compression results of trace headers using four different entropy coding methods:

- Two Huffman coding (HC) schemes:
  1. non-conditional semi-adaptive HC;
  2. non-conditional adaptive HC;

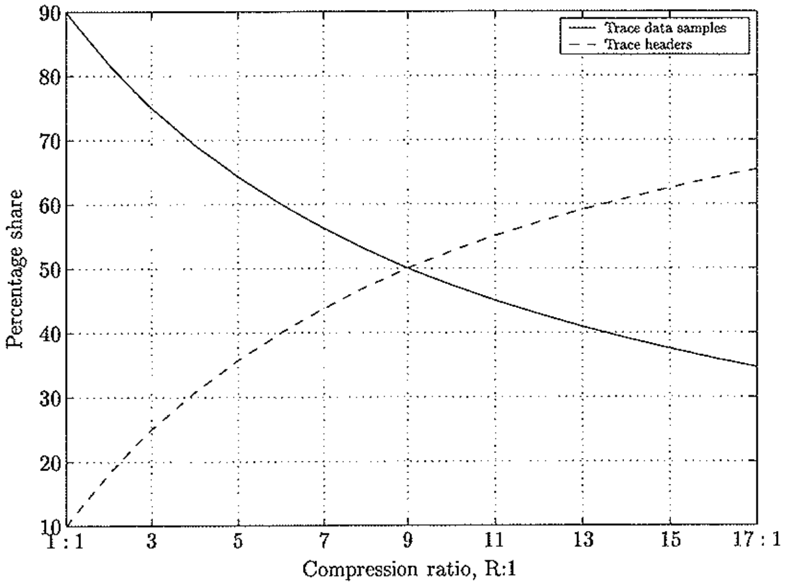


Figure 3.1: Percentage share of trace data samples and trace headers versus compression ratio of trace data samples.

- Two arithmetic coding (AC) schemes:

   3. non-conditional semi-adaptive AC;
   4. conditional adaptive AC.

HC and AC are two of the most well-known entropy coding methods. Their principles are given in Appendix 3.A. To compress data using HC or AC, we need a statistical model of the letters to be compressed. Accurate estimation of the probabilities of symbols in the input data is an inherent need in HC and AC. A zero-order or non-conditional probability model is less complex than a higher-order or conditional probability model. The conventional way of data compression is first to make a pass over the letters to be compressed to gather statistics for the probability model. Then a second pass is made to actually compress the data. The statistics are normally sent as side information so that the decoder will have a copy. Such a semi-adaptive entropy coding approach obviously has serious problems if the statistics of the probability model take more space than the data to be compressed. This situation likely occurs for complicated data and/or complex probability models. Adaptive entropy coding is the solution to this dilemma. In adaptive entropy coding, both the encoder and the decoder start with the same probability model and gradually update the statistics. Adaptive entropy coding has a slight disadvantage in that it starts the compression with non-optimal statistics. Given the cost of transmitting the statistics as side information, an adaptive entropy coding algorithm usually performs better than a semi-adaptive entropy coding technique. In general, a conditional probability model requires an adaptive entropy coding method.

This paper is organized as follows. First, the trace identification headers of the SEG-Y standard are characterized. Next, we present the lossless compression algorithm for the trace identification headers based on differential coding and run-length coding in conjunction with class-wise entropy coding of symbols (i.e., the run) and counts (i.e., the length). Subsequently the trace identification headers of common offset gathers from a 2-D seismic data set are compressed using the four different entropy coding methods. In the results we attend the dependency of the compression performance as a function of the order of the conditional probability model applied in the adaptive AC scheme. Finally, discussion and conclusions end the paper.

This article is the second in a collection of three papers on seismic data compression. Lossy compression of seismic data (i.e., the trace data samples) based on subband coding (SBC) is presented in Part I (Røsten et al., 1999b) while an analysis of the influence of seismic data compression on processing and interpretation, using the proposed SBC, is presented in Part III (Røsten et al., 2000b).

3.3 Seismic data format standards

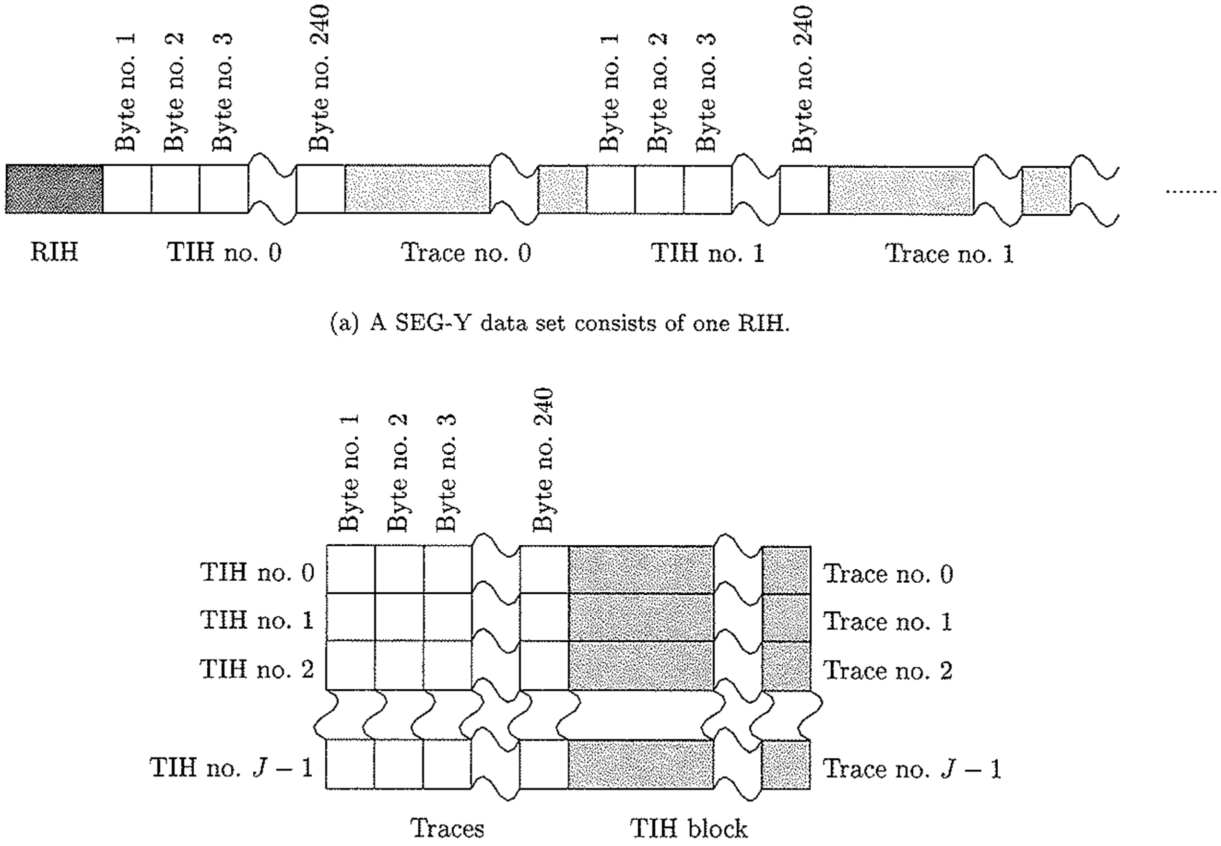
During the early years of digital recording of seismic data, numerous recording formats were put into use. The resulting proliferation problem led to the development of the SEG-A, the SEG-B, and the SEG-X seismic data format standards that were introduced by the Society of Exploration Geophysicists (SEG) in 1967 (Northwood et al., 1967). In 1972, the SEG-C standard (Meiners et al., 1972) was introduced by the SEG due to several new developments in field system technology. The SEG-Y standard was developed by the SEG in 1975 (Barry et al., 1975) for the extended use of workstations in field systems. The growing use of multiple streamers in seismic data acquisition governed the SEG-D standard published by the SEG in 1977 (see e.g., SEG Tech. Std. Comm., 1997). Later the SEG-D has been revised two times, in 1994 and in 1996, to increase its flexibility and ability to manage ocean bottom seismic. Today, the trend is that the SEG-Y and the SEG-D standards are the ones used during processing and field recordings of seismic data, respectively. For clarity, we focus on the SEG-Y standard.

3.3.1 The SEG-Y standard

A SEG-Y data set is divided into three parts [see Figure 3.2 (a)]; the reel identification header (RIH), the trace identification headers (TIHs), and the trace data samples. We assume that the native workstation word format is utilized for all three parts. The use of the native format will not affect the compression performance, but is, strictly speaking, sometimes a misuse of the SEG-Y standard. For instance, the legal word format of the trace data samples is one of four different IBM formats, while the 32-bit IEEE floating-point number representation is the native format on many workstations (e.g., SGI and SUN).

The RIH contains 3600 bytes of information used to identify the entire data set and is subdivided into two sub-blocks. The first sub-block is the Extended Binary Coded Decimal Interchange Code (EBCDIC) card image block organized into 40 card images with 80 byte EBCDIC each (i.e., 40 lines of text with 80 characters per line). Card image line numbers 23 through 39 are unassigned for optional use. The second sub-block of the RIH is the binary coded section of 400 bytes of information valid for the full data set. The leading 60 bytes in this second sub-block are assigned while the trailing 340 bytes are unassigned for optional use. The size of the RIH is negligible compared to a typical size of trace data samples. Consequently, we do not compress the RIH in this work [see Figure 3.2 (b)].

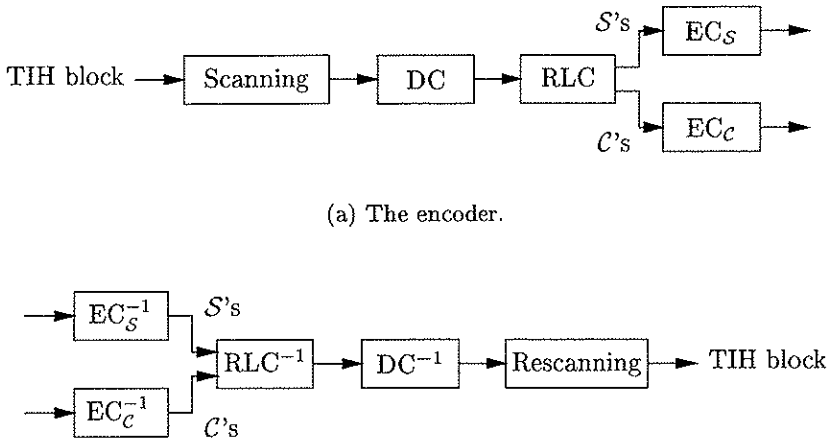
Each trace is preceded by a 240 byte TIH. The leading 180 bytes are assigned and the trailing 60 bytes are unassigned for optional use. The leading 180 assigned bytes are grouped into letters of either two or four bytes when interpreted. The TIH block is used to specify the characteristics of the recording, the geometry, the preprocessing parameters, and so forth. For a trace containing 1001 data samples (e.g., 4000 ms of trace data samples at a sampling interval equal to 4 ms), the total size of the TIHs is 6 % of the size of the trace data block. As a rule of thumb, we say that 10 % of a SEG-Y data set is composed of TIHs and 90 % of a SEG-Y data set is consequently trace data samples (the RIH is ignored). Therefore, if the TIH block is maximally compressed $\tilde{R}:1$, then compression ratios $R:1 > 9\times\tilde{R}:1$ for the trace data samples provide less marginal coding efficiency. Note, a compression ratio of $\tilde{R}:1$ of the trace identification headers corresponds to $8/\tilde{R}$ bits per byte number. Similarly, a compression ratio of $R:1$ of the trace data samples



(a) A SEG-Y data set consists of one RIH.

(b) The J numbers of TIHs and the associated traces as a matrix.

Figure 3.2: The SEG-Y standard seismic data format.



(a) The encoder.

(b) The decoder.

Figure 3.3: The lossless coding system for TIHs.

corresponds to $32/R$ bits per sample.

3.4 Methodology

In Røsten et al. (1999b), we presented a lossy SBC scheme for the trace data samples. Here, we propose an effective lossless compression algorithm for the trace identification headers based on entropy coding (see e.g., Røsten and Ramstad, 1999; Røsten et al., 2000a).

3.4.1 Characteristics of the TIHs

In general, two main characteristics of the TIH block are recognized and utilized by the proposed compression algorithm. First, some of the bytes are recognized to be constant across many traces. For instance, the TIH entries "Water depth at source" and "Water depth at group", designated in byte numbers 61 through 64 and 65 through 68, respectively, are usually constant for all the traces. Second, the values for some of the byte numbers are recognized to increase or decrease by a constant value when moving from trace to trace. This condition is frequently satisfied, for instance, for the TIH entry "Trace sequence number within reel" specified in bytes 5 to 8. In addition, some of the bytes obey both characteristics. For example, the TIH entry "Energy source point

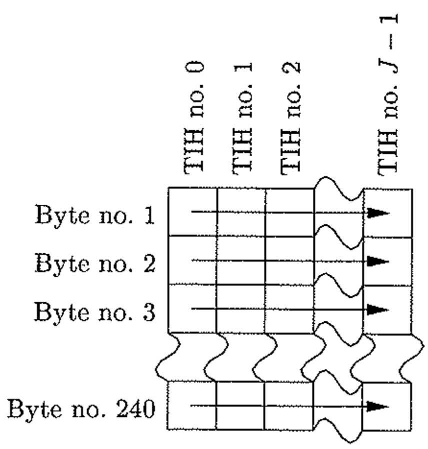


Figure 3.4: Scanning of the TIH block. The arrows indicate the row-wise byte-plane scanning (across the direction of the traces) utilized before differential coding and run-length coding.

number" described in bytes 17 to 20, is usually constant for all the traces which belong to the same shot, but is normally increased by one when sequentially moving from shot to shot.

3.4.2 Structure of the lossless coding system

The lossless coding system for trace identification headers is sketched in Figure 3.3. First, at the encoder [see Figure 3.3 (a)], the TIH block is scanned byte-wise across the traces, starting with byte number $b = 1$ and ending with byte number $b = 240$ (see Figure 3.4). The byte-wise scanning reshapes a 2-D matrix to a 1-D vector, organizes the header data in a natural manner for compression, simplifies the reading of the variable-length letters while, finally, preserves the two postulated characteristics of the TIH block. Second, 1-D first-order fixed differential coding (DC) is utilized to decrease the redundancy between the bytes after the scanning procedure (i.e., 1-D). First-order fixed DC consists of a single constant prediction coefficient in an error prediction filter. It is very common that subsequent bytes are equal or differ with one in absolute value, and for this reason, we assign a prediction coefficient equal to 1.0 in the DC scheme (see Figure 3.5). Third, we utilize run-length coding (RLC) to take advantage of bytes from the error prediction filter that are constant across many traces. RLC is a simple technique used to compress runs of identical symbols in a data stream. It encodes a run of symbols as a symbol ($\mathcal{S}$) and a count ($\mathcal{C}$) which represents the number of symbols in the run. The $\mathcal{C}$'s

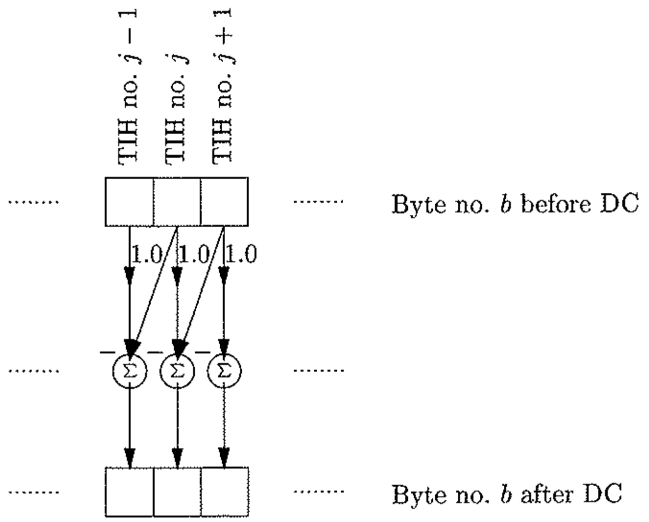


Figure 3.5: Configuration of 1-D first-order fixed DC with a prediction coefficient equal to 1.0. The figure shows byte no. b for $1 \leq b \leq 240$ between TIH no. $j-1$ and TIH no. $j+1$ for $2 \leq j \leq J-2$.

for all the $\mathcal{S}$'s are in principle increased by the byte-wise scanning and the DC. As the core of the lossless coding system, we use entropy coding (EC). Since the RLC gives two data sequences with different probability distributions, we use one EC scheme for the $\mathcal{S}$'s (denoted $\text{EC}_\mathcal{S}$) and another EC scheme for the $\mathcal{C}$'s (denoted $\text{EC}_\mathcal{C}$), that is class-wise entropy coding.

The decoder [see Figure 3.3 (b)] performs in general the opposite operations in reverse order: entropy decoding, denoted $\text{EC}_\mathcal{S}^{-1}$ and $\text{EC}_\mathcal{C}^{-1}$, respectively, to reconstruct the $\mathcal{S}$'s and the $\mathcal{C}$'s. The $\mathcal{S}$'s and the $\mathcal{C}$'s are then passed through inverse RLC, RLC^{-1}, and 1-D first-order fixed inverse DC, DC^{-1}. The TIH block follows after rescanning at the end.

3.4.3 Probability models for entropy coding

The idea of EC is to assign short bit-codes to letters that appear frequently while reserving long bit-codes to less frequent ones (i.e., variable-rate coding). We compare HC and AC, two of the most well-known EC methods. In general, AC offers higher compression ratios than HC (see Appendix 3.A), but AC is more complex to implement compared to HC. In both cases, a good estimate of the probability density function for the letters to be compressed is needed for efficient compression.

3.4.3.1 Non-conditional probability model

In Røsten et al. (2000a), a simple non-conditional or zero-order probability model was found to give satisfactory compression results. Assume for simplicity that we denote the sequences of $\mathcal{S}$'s and $\mathcal{C}$'s by the letter $s(k)$ for $k = 0, 1, \ldots, K-1$. Furthermore, suppose that $s(k)$ is one out of $M \leq K$ distinct values or symbols $s(k) \in \mathcal{A}_m$ for $m = 0, 1, \ldots, M-1$. Then the non-conditional probability $P_S(s)$ (the sequence of letters are represented by a stochastic variable S) is determined from:

$$P_S[s(k)] = P_S(\mathcal{A}_m) = \frac{\mathcal{K}(\mathcal{A}_m)}{K}, \tag{3.1}$$

where $\mathcal{K}(\mathcal{A}_m)$ denotes the number of occurrences of symbol $\mathcal{A}_m$ such that

$$\sum_{m=0}^{M-1} \mathcal{K}(\mathcal{A}_m) = K. \tag{3.2}$$

3.4.3.2 Conditional probability model

Nevertheless, neither the $\mathcal{S}$'s nor the $\mathcal{C}$'s are necessarily memoryless. In fact, in Røsten and Ramstad (1999) it was recognized that the $\mathcal{S}$'s and the $\mathcal{C}$'s, despite the DC and the RLC, typically and locally seem to have a kind of periodicity. Hence, we can use a conditional probability model of order $\mathcal{O}$, $P_S^{\mathcal{O}}(s)$, instead of the zero-order probability model, $P_S(s)$:

$$P_S^{\mathcal{O}}[s(k)] = P_S[s(k) \mid s(k-1), s(k-2), \ldots, s(k-\mathcal{O})]. \tag{3.3}$$

3.5 Compression results

The proposed lossless compression algorithm is evaluated for the trace identification headers belonging to the 240 common offset gathers from the 2-D seismic data set we used in Røsten et al. (2000b). Notice that the RIH and the trace data samples are stripped off and ignored for the purpose of this paper. The following four multi-alphabet EC schemes are compared (i.e., EC$_{\mathcal{S}}$ and EC$_{\mathcal{C}}$ are subsequently one of these four types):

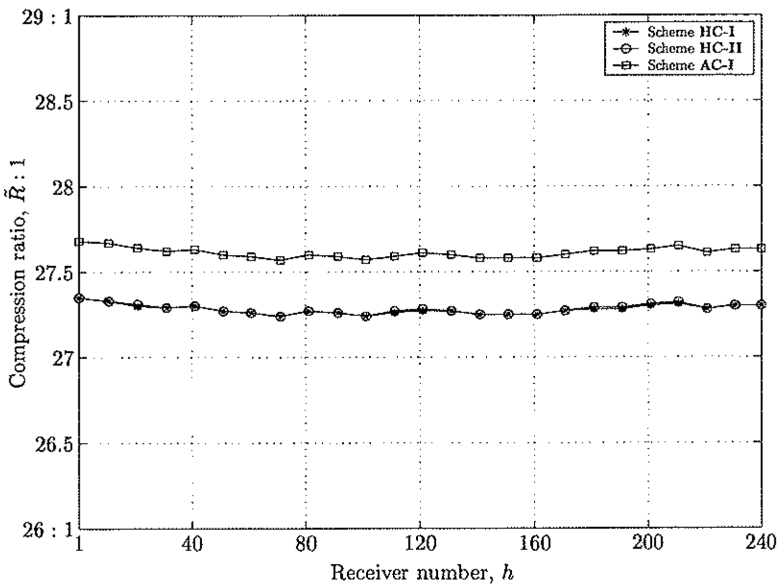


Figure 3.6: $\tilde{R}$: 1 versus offset; the three non-conditional EC methods.

- Scheme **HC-I**:

  1. $\mathrm{EC}_{\mathcal{S},\mathcal{C}}$ are non-conditional semi-adaptive HC systems;

- Scheme **HC-II**:

  2. $\mathrm{EC}_{\mathcal{S},\mathcal{C}}$ are non-conditional adaptive HC systems;

- Scheme **AC-I**:

  3. $\mathrm{EC}_{\mathcal{S},\mathcal{C}}$ are non-conditional semi-adaptive AC systems;

- Scheme **AC-II**:

  4. ${}^{\mathcal{O}}\mathrm{EC}_{\mathcal{S},\mathcal{C}}$ are conditional adaptive AC systems of order $\mathcal{O}$.

Figure 3.6 displays the compression ratio (CR) versus receiver number, h, for schemes **HC-I**, **HC-II**, and **AC-I**. The CR is simply defined as the size of the trace identification headers before compression, divided by the size of the trace identification headers after compression. As previously mentioned, a CR of $\tilde{R}$: 1 corresponds to $8/\tilde{R}$ bits per byte number. As can be seen, the non-conditional semi-adaptive AC method performs, as expected, slightly better than the two HC approaches. Non-conditional semi-adaptive HC and non-conditional adaptive HC have more or less similar compression ratios.

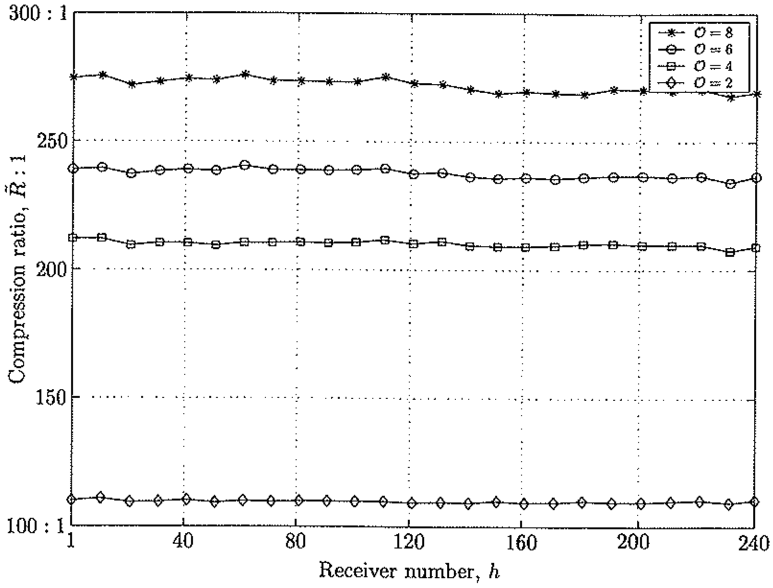


Figure 3.7: $\tilde{R} : 1$ versus offset; conditional adaptive AC for $\mathcal{O} = \{2, 4, 6, 8\}$.

Figure 3.7 displays the CR versus receiver number for scheme **AC-II**, that is the most complex EC method, using orders of the conditional probability model ranging from $\mathcal{O} = 2$ to $\mathcal{O} = 8$. Remark, the order is always the same for ${}^{\mathcal{O}}\text{EC}_{\mathcal{S}}$ and ${}^{\mathcal{O}}\text{EC}_{\mathcal{C}}$. The resulting compression ratio is monotonically increasing as a function of the order up to $\mathcal{O} = 8$, but $\mathcal{O} = 9$ and $\mathcal{O} = 10$ represent oversized orders which somewhat reduce the compression ratios (see Figure 3.8).

Most interesting, despite the "mile-high" improved coding results for scheme **AC-II**, $\tilde{R} : 1$ is without exception so high for all four tested EC methods that the size of the trace identification headers becomes, independently of the applied EC method, negligible compared to the size of the associated trace data samples.

3.6 Discussion

A straightforward implementation of the conditional adaptive AC algorithm results in a highly increasing memory consumption as $\mathcal{O}$ is enlarged; if the order of the conditional probability model increases linearly, then the memory consumed by the conditional probability model increases exponentially (Nelson and Gailly, 1996). Fortunately, there is a solution to this problem. Instead of letting every value or symbol to appear automatically in the probability table,

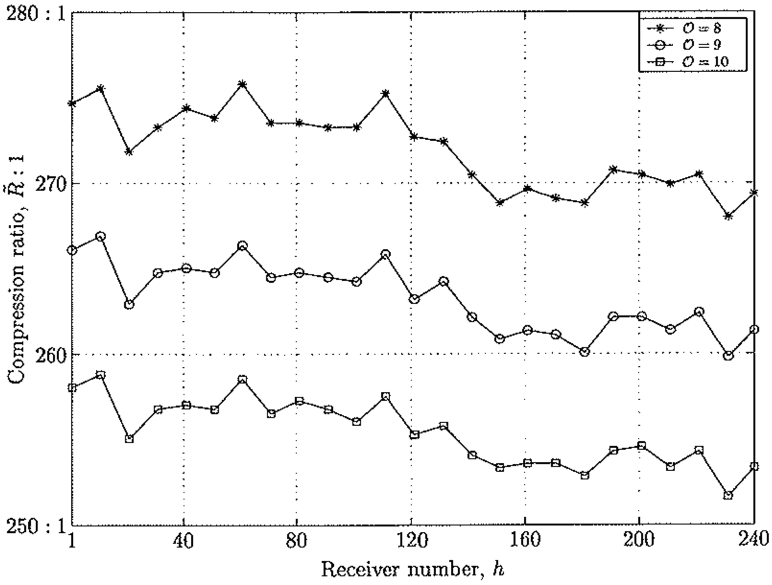


Figure 3.8: $\tilde{R} : 1$ versus offset; conditional adaptive AC for $\mathcal{O} = \{8, 9, 10\}$.

we start off with an empty probability table and add values or symbols to the probability table only as they appear. In addition, the probability table is organized as a tree. This organization lets the conditional model adjust rapidly and efficiently to changing statistics. Tests indicate that with this enhanced conditional adaptive AC algorithm, the CPU time becomes approximately independent of the order. Hence, based on the compression results, a suitable order in the case of common offset gathers is 8. This order is also recommended for common shot gathers (Røsten and Ramstad, 1999).

Since the RIH is not compressed, the unassigned portion of it can be utilized to memorize parameters used in the compression of the SEG-Y data set for later decompression.

3.7 Conclusions

The use of conditional adaptive arithmetic coding is highly successful in lossless compression of the trace identification headers of the SEG-Y standard. Compression ratios up to 275:1 have been demonstrated in the case of common offset gathers, the optimal order of the conditional probability model being 8. It is trivial to extend the compression algorithm to other seismic data format standards than the SEG-Y. The proposed compression technique is computationally fast. Such compression can therefore be incorporated as a natural component in the storage and transmission of seismic data.

3.8 Acknowledgments

Tage Røsten thanks Den norske stats oljeselskap a.s (Statoil) and Petroleum Geo-Services (PGS) for financial support. The authors thank Dr. Gijs Vermeer for his constructive comments.

3.A Entropy coding examples

We give numerical examples of multi-alphabet HC and AC (see e.g., Gersho and Gray, 1991; Nelson and Gailly, 1996) for a test sequence. We assume for simplicity that the input letter $s(k)$ for $k = 0, 1, \ldots, K-1$ is one out of four distinct symbols $s(k) \in \mathcal{A}_m$ for $m = 0, 1, \ldots, 3$. The following five-symbol (i.e., $K = 5$) sequence or message, $s(0)s(1)s(2)s(3)s(4) = \mathcal{A}_0\mathcal{A}_1\mathcal{A}_2\mathcal{A}_2\mathcal{A}_3$, is compressed by the two variable-rate coding techniques. We have a zero-order or non-conditional probability model where the probability of occurrence $P_S(s)$ is given by

$$P_S(s) = \begin{cases} 0.2, & s(k) = \mathcal{A}_0 \\ 0.2, & s(k) = \mathcal{A}_1 \\ 0.4, & s(k) = \mathcal{A}_2 \\ 0.2, & s(k) = \mathcal{A}_3. \end{cases} \tag{3.A-1}$$

Moreover, the probability model is predetermined and fixed (i.e., neither semi-adaptive nor adaptive). The probabilities are known at the encoder and the decoder, and the side information associated with the probability model is thus neglected.

3.A.1 Non-conditional non-adaptive Huffman coding

The first step in HC is to create a series of source reductions by ordering the probabilities of the letters under consideration, and combining the two lowest probability symbols into a single symbol that replaces them in the next source reduction step. Table 3.A-1 illustrates the first step in HC. At the far left, the set of (source) symbols and their probabilities are ordered from top to bottom in terms of decreasing probability value. To form the first source reduction, the two bottom probabilities marked with a red box, 0.2 and 0.2, are combined to a "compound symbol" with probability 0.4. This compound symbol and its associated probability (written in yellow text) are placed in the first source reduction column so that the probabilities of the reduced source are also ordered from the most to the least probable. This source assignment process is repeated until a reduced source with two symbols (at the far right) is reached.

The second step in HC is to encode each reduced source, starting with the smallest reduced source and working back to the original source. The minimal length binary code for a two-symbol source is 0 and 1. As Table 3.A-2 shows, these codes (written in blue text) are assigned to the two symbols on

Table 3.A-1: Source reduction step in HC, going from left to right.

Original source		Source reduction		
Symbol	Probability	0.	1.	2.
$\mathcal{A}_2$	0.4	0.4	0.4	0.6
$\mathcal{A}_0$	0.2	0.2	0.4	0.4
$\mathcal{A}_1$	0.2	0.2	0.2	
$\mathcal{A}_3$	0.2	0.2		

Table 3.A-2: Source assignment step in HC, going from right to left.

Original source			Source reduction		
Symbol	Probability	Code	0.	1.	2.
$\mathcal{A}_2$	0.4	1	0.4 1	0.4 1	0.6 0
$\mathcal{A}_0$	0.2	01	0.2 01	0.4 00	0.4 1
$\mathcal{A}_1$	0.2	000	0.2 000	0.2 01	
$\mathcal{A}_3$	0.2	001	0.2 001		

the right marked with a yellow box (the assignment is arbitrary; reversing the order of 0 and 1 would work just as well). As the reduced source symbol with probability 0.6 (written in red text) was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and 0 and 1 are arbitrarily appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original source is reached. The final binary code appears in the third column of Table 3.A-2. In general, short bit-codes are assigned to symbols that appear frequently while long bit-codes are reserved to less frequent ones. The average length of this message is

$$\bar{L} = 0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.2 \times 3 = 2.0 \text{ bits per symbol,} \qquad (3.A\text{-}2)$$

while the Shannon or first-order entropy of the source is

$$H = -\sum_{m=0}^{3} P_S(\mathcal{A}_m) \log_2[P_S(\mathcal{A}_m)] \approx 1.92 \qquad (3.A\text{-}3)$$

bits per symbol. Thus the coding efficiency of the Huffman code for the test sequence is 96 %. Remark, this example is somewhat artificial since fixed-rate coding of the message also gives 2.0 bits per symbol, i.e., equal to the average length of the message provided by the HC method.

The Huffman code is the optimal code for a set of symbols and probabilities subject to the constraint that the symbols are coded one at a time. After the code has been created, encoding and/or decoding is accomplished in a simple table look-up manner. The code itself is instantaneous, if starting at the correct bit position, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. In addition, it is uniquely decodable because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left-to-right manner. For the binary code given in Table 3.A-2, a left-to-right scan of the 10-bit encoded string 0100011001 reveals that the first valid code word is 01, which is the code for symbol $\mathcal{A}_0$. The next valid code word is 000, which corresponds to symbol $\mathcal{A}_1$. Continuing in this manner reveals the completely decoded message to be $\mathcal{A}_0\mathcal{A}_1\mathcal{A}_2\mathcal{A}_2\mathcal{A}_3$.

3.A.2 Non-conditional non-adaptive arithmetic coding

Unlike HC, AC generates non-block codes. In AC, a one-to-one correspondence between source symbols and code words does not exist. Hence, AC is a non-instantaneous type of entropy coding. Instead, an entire sequence or message of input letters is assigned a single arithmetic code word. The code word itself defines a half-open interval of real numbers greater than or equal to 0.0 and less than 1.0. As the number of input letters in the message increases, the interval used to represent it becomes smaller and the number of bits required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with is probability of occurrence. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the encoded string. Because this technique does not require, as does traditional HC, that each source symbol translates into an integral number of bits (that is, the symbols are coded one at at time) AC achieves (but only in theory) the bound established by the Shannon or first-order entropy. Note that more advanced HC methods can also reach the Shannon bound by grouping together several original symbols to new symbols.

As already stated, the output from an AC process is a single number greater than or equal to $P_b = 0.0$ and less than $P_e = 1.0$. This single number can be uniquely decoded to create the exact stream of symbols that went into its construction. At the start of the AC process, the message is assumed to occupy the entire half-open interval $[P_b, P_e) = [0.0, 1.0)$. In our case, this interval is initially subdivided into four regions based on the probabilities of each source symbol (see Table 3.A-3). Symbol $\mathcal{A}_0$ for example, is associated

Table 3.A-3: Initial subintervals in arithmetic coding.

Original source		
Symbol	Probability	Initial subinterval
$\mathcal{A}_0$	0.2	[0.0,0.2)
$\mathcal{A}_1$	0.2	[0.2,0.4)
$\mathcal{A}_2$	0.4	[0.4,0.8)
$\mathcal{A}_3$	0.2	[0.8,1.0)

with subinterval $[0.0, 0.2)$. Because it is the first symbol of the message being coded, the message interval is at the start narrowed to $[0.0, 0.2)$. Specifically, if $\mathcal{A}_j$ for $0 \le j \le 3$ is the present symbol to be coded, then the message interval is narrowed according to

$$\hat{P}_b = P_b + \left[\sum_{m=0}^{j-1} P_S(\mathcal{A}_m)\right] \times (P_e - P_b) \tag{3.A-4a}$$

and

$$\hat{P}_e = P_b + \left[\sum_{m=0}^{j} P_S(\mathcal{A}_m)\right] \times (P_e - P_b), \tag{3.A-4b}$$

with $P_e = \hat{P}_e$ and $P_b = \hat{P}_b$ after each encoded symbol.

The probability range is continuously narrowed according to equations 3.A-4a and 3.A-4b for each source symbol, until all symbols in the message are put into the interval narrowing process. In this manner, the second symbol $\mathcal{A}_1$ narrows the subinterval to $[0.04, 0.08)$, the third symbol $\mathcal{A}_2$ further narrows it to $[0.056, 0.072)$, and so on (see Table 3.A-4). The final message symbol $\mathcal{A}_3$ narrows the range to $[0.06752, 0.0688)$. Of course, any number within this subinterval, for example 0.068, can be used to represent the message.

In the arithmetically coded message of Table 3.A-4, minimum three decimal digits are used to represent the five-symbol message. This translates into

$$\bar{L} = 3/5 = 0.6 \text{ decimal digits per symbol} \tag{3.A-5}$$

using base 10, since the above encoding was performed in decimal format. The Shannon or first-order entropy of the source is

$$H = -\sum_{m=0}^{3} P_S(\mathcal{A}_m)\log_{10}[P_S(\mathcal{A}_m)] \approx 0.58 \tag{3.A-6}$$

Table 3.A-4: Arithmetic encoding example.

Message	Probability range $[P_b, P_e)$
$\mathcal{A}_0$	[0.0,0.2)
$\mathcal{A}_1$	[0.04,0.08)
$\mathcal{A}_2$	[0.056,0.072)
$\mathcal{A}_2$	[0.0624,0.0688)
$\mathcal{A}_3$	[0.06752,0.0688)

Table 3.A-5: Arithmetic decoding example.

Encoded information		Probabilities	
Number	Output symbol	Initial subinterval	Distance
0.068	$\mathcal{A}_0$	[0.0,0.2)	0.2
0.34	$\mathcal{A}_1$	[0.2,0.4)	0.2
0.7	$\mathcal{A}_2$	[0.4,0.8)	0.4
0.75	$\mathcal{A}_2$	[0.4,0.8)	0.4
0.875	$\mathcal{A}_3$	[0.8,1.0)	0.2

decimal digits per symbol. Thus the coding efficiency of the arithmetic code for the test sequence is 96.7 %, which is slightly higher than the coding efficiency of the Huffman code. Of course, we normally work in binary format, transmitting bits and measuring entropy in bits per symbol.

In practice, two factors cause AC performance to fall short of the bound. First, the message-length K (or alternatively a special end-of-file indicator) must be appended to the encoded string to separate one message from another. Second, the use of finite precision arithmetic somewhat complicates the AC process (see e.g., Nelson and Gailly, 1996). Practical implementations of AC address the latter problem by introducing a scaling strategy and a rounding strategy (Rissanen and Langdon, 1981; Witten et al., 1987). The scaling strategy renormalizes each subinterval to the $[0.0, 1.0)$ probability range before subdividing it in accordance with the symbol probabilities. The rounding strategy guarantees that the truncations associated with the use of finite precision arithmetic do not prevent the coding subintervals from being represented accurately.

Given the encoding example in Table 3.A-4, it is trivial to see how the decoding process operates (see Table 3.A-5). Find the first symbol in the message by seeing which symbol owns the space our encoded string falls in. Since 0.068 falls between 0.0 and 0.2, the first character must be $\mathcal{A}_0$. Then remove $\mathcal{A}_0$ from the encoded number; since we know the lower and upper limit of the probability range of $\mathcal{A}_0$, remove its effects by reversing the encoding process. First subtract the lower limit of $\mathcal{A}_0$, giving 0.068 again. Second divide by the width of the range of $\mathcal{A}_0$, or 0.2. This gives 0.34. Then locate in which interval that number falls, which is in the range of the next letter in the message, $\mathcal{A}_1$. Continuing in this manner reveals again the completely decoded message to be $\mathcal{A}_0\mathcal{A}_1\mathcal{A}_2\mathcal{A}_2\mathcal{A}_3$.

3.A.3 On the coding efficiency of HC and AC

AC has significantly higher coding efficiency compared to HC when a symbol has probability of occurrence approaching unity (see e.g., Nelson and Gailly, 1996), for example 0.9, and in the case of long messages, i.e., K much greater than 5. A symbol emanating from such a source conveys negligible information (≈ 0.15 bits per symbol), but requires at least 1 bit to be transmitted in the case of conventional HC. AC dispenses with the restriction that each symbol must translate into an integral number of bits, thereby coding more efficiently. A well suited probability model (e.g., conditional) exposes the deficiencies of traditional HC compared to AC more starkly than non-optimal ones (Witten et al., 1987). This is because advanced probability models more commonly predict symbols with probability close to one, the worst case for HC. Thus, *conditional adaptive* HC is not an appropriate entropy coding method.

Paper 3

Chapter 4

Seismic data compression, Part III: Its influence on processing and interpretation

4.1 Abstract

We investigate the impact of lossy seismic data compression on processing and interpretation of a 2-D seismic data set acquired in the North Sea. The seismic data set is sorted into common offset gathers which are separately compressed using a 2-D subband coding method. The associated trace headers are compressed by a lossless arithmetic coding technique. After compression and decompression, we study the influence of the introduced coding noise to a representative processing sequence. Important processing steps, in chronological order, include attenuation of water-bottom multiples, true amplitude recovery, predictive deconvolution, attenuation of peg-leg multiples, migration, and stacking. In general, migration and stacking reduce the effect of the coding noise at all compression levels, while for high compression ratios (i.e., much greater than 10:1) the coding noise has a destructive effect on the predictive deconvolution step and on the two applied multiple attenuation methods. Two sequences of the lossy compression method are explored. In the first sequence the seismic data are compressed *before* predictive deconvolution (denoted pre-decon compression), while in the second sequence the seismic data are compressed *after* predictive deconvolution (denoted post-decon compression). Several seismic data responses are examined, for example prestack and poststack amplitude analysis in addition to poststack inversion. For pre-decon

and post-decon compression, compression ratios between 7.5:1-15:1 and 15:1-30:1, respectively, provide excellent reconstruction quality of the seismic data set.

4.2 Introduction

Data compression makes the storage of seismic data more efficient, and has a potential of reducing the time for network and satellite transmission of seismic data from hours to minutes. Hence, a significant reduction in cost is achieved. For example, a compression ratio of 20:1 reduces one terabyte of seismic data to 50 gigabytes. Taking the cost of storage per megabyte of seismic data on robotic tape to be $0.20 per year, the storage price would be reduced by 190,000 dollars per year per terabyte of raw data. Furthermore, assuming a satellite transmission cost equal to $2.0 per megabyte of seismic data, the transmission cost of the same compression example would be decreased from 2.0 million dollars to 100,000 dollars per terabyte of raw data.

Two main types of compression techniques exist; lossless and lossy compression. No errors are introduced into the data by lossless compression and the seismic data set can be reconstructed without any loss of information. The least significant bits (containing mostly ambient noise) and the most significant bits in this case are given equal weight of importance. Since all bits are reconstructed exactly, lossless compression of seismic data can only offer compression ratios in the order of 2:1. Compression of text files (e.g., trace identification headers), on the other hand, can achieve compression ratios in the order of several hundreds to one by lossless methods (Røsten et al., 1999c). This is because a text file contains frequently repeated letters and patterns of letters while a typical seismic data file is far less structured (Reiter and Heller, 1994). During lossy compression of seismic data some errors are introduced. Due to the introduction of compression noise, lossy methods generally offer higher compression ratios than lossless methods. The characteristics of the compression noise depend on the compression ratio. A high compression ratio (much higher than 10:1) gives a high level of compression noise. At the same time, the compression noise becomes more and more coherent (i.e., non-white) as the compression ratio increases.

Throughout all parts of the seismic industry, from acquisition to processing and interpretation, users are careful not to employ procedures that introduce noise into the seismic data set. Nevertheless, as such, lossy seismic data compression can be seen as just one of the noise sources in the acquisition and processing of seismic data. For example, only 1 % perturbation in the stacking velocity field can lead to very different processing results (Bosman and Reiter, 1993). More int erestingly, lossy compression actually removes some of the ambient noise inherent in the seismic data. This observation has been stressed by Dessing and Hoekstra (1997). It is not necessarily a problem to introduce distortion while carrying out certain processing steps. However, it is a problem if the properties of the noise are unknown, in addition to an ignorance of the noise's influence on the seismic data. In light of this uncertainty, Vermeer (1999) suggested that the compression distortion should not affect the dynamic range of the overall acquisition system, describing lossless compression "in all but name". Vermeer et al. (1996) and Donoho et al. (1999b), being less restrictive than Vermeer (1999), proposed that the compression distortion should not exceed the level of, and should have the same characteristics (e.g., additive white Gaussian noise) as other (i.e., ambient) noise components. Being even less restrictive than Vermeer et al. (1996) and Donoho et al. (1999b), we seek to explore some of these uncertainties by not constraining the compression noise in any way. With an understanding of the associated drawbacks we expect that the potential of reduced turnaround times, from seismic data acquisition to processing and interpretation, will push the use of lossy compression as a routine component of standard survey operations.

Traditionally, the size of seismic data is reduced by for example reduction of the temporal range and/or resampling in time, reduction of the spatial range and/or resampling in space, and clipping of the amplitudes, thereby reducing the dynamic range. Seismic data acquisition usually provides 2.0 ms and 12.5 m sampling intervals in time and space, respectively. Still, before processing, the seismic data are commonly down-sampled by a factor of two both temporally and spatially (see e.g., Stigant et al., 1995). This down-sampling implies a compression ratio of 4:1 for 2-D seismic data, and is strictly speaking a lossy process since seismic data are non-band-limited in time and space. Seismic data are originally represented by 32 bits per sample (floating-point number representation). For many purposes, however, a fixed-point number representation of 16 bits per sample provides sufficient numeric precision. Hence, seismic data are further lossy compressed by 2:1.

Different from such conservative approaches, we consider lossy seismic data compression as a more sophisticated alternative than traditional seismic data reduction methods; lossy compression preserves the dynamic range of the seismic signal and automatically removes "unnecessary" seismic data components at high frequencies and high wavenumbers.

Previous work in lossy seismic data compression are mainly based on so-called frequency domain coders. Subband coding (SBC; see e.g., Røsten et al., 1996, 1997a), discrete wavelet transform coding (DWTC; see e.g., Luo and Schuster, 1992; Reiter and Heller, 1994; Donoho et al., 1995), and transform coding (TC; see e.g., Spanias et al., 1991; Vermeer et al., 1996; Duval et al., 1999b) belong all to frequency domain coders. The origin of frequency domain based compression methods was within the field of speech (Crochiere et al., 1976) and image coding (Woods and O'Neil, 1986). Speech, audio, image, and video coding have SBC applications for storage and transmission, and the compression ratio may be set as high as desired, consistent with demands of achieved quality after decompression put forward by human perception. If further data processing is required, the processing is most conveniently carried out with the original uncompressed data. Compression of seismic data, however, is more complicated. For example, if compression is applied in the acquisition stage offshore, and the compressed seismic data set is transmitted to land, the effects of compression noise on the seismic data must be negligible since the complete processing flow is carried out with the seismic data after decompression.

A successful field trial using satellite communication from offshore to land in combination with a 4-D lossy seismic data compression scheme has been reported by Stigant et al. (1995). Preliminary results showed that a compression ratio of 50:1 preserved the quality of the seismic data (Stigant et al., 1995). Unfortunately, no definite conclusions on the character of the compression noise and its influence on the seismic data have been reported. An industry-wide consortium designated the Seismic Compression Diagnostic Initiative (SCDI) was formed in 1997 to address some of these questions (see e.g., Donoho et al., 1999a,b, 2000).

The seismic data organization (i.e., dimensionality and sorting) is important to achieve large compression ratios with an acceptable distortion level. For instance, 2-D seismic data compression is in general more efficient than 1-D seismic data compression due to decreased redundancy in the latter case. A quantitative comparison of 1-D, 2-D, and 3-D seismic data compression using DWTC has been done by Reiter (1996). Moreover, common offset gathers seem to be more suited for compression than common shot gathers (Dessing and Hoekstra, 1997; Røsten et al., 1999a). For example, primary reflection events are horizontally aligned for a non-dipping surface in the former case, giving higher redundancy.

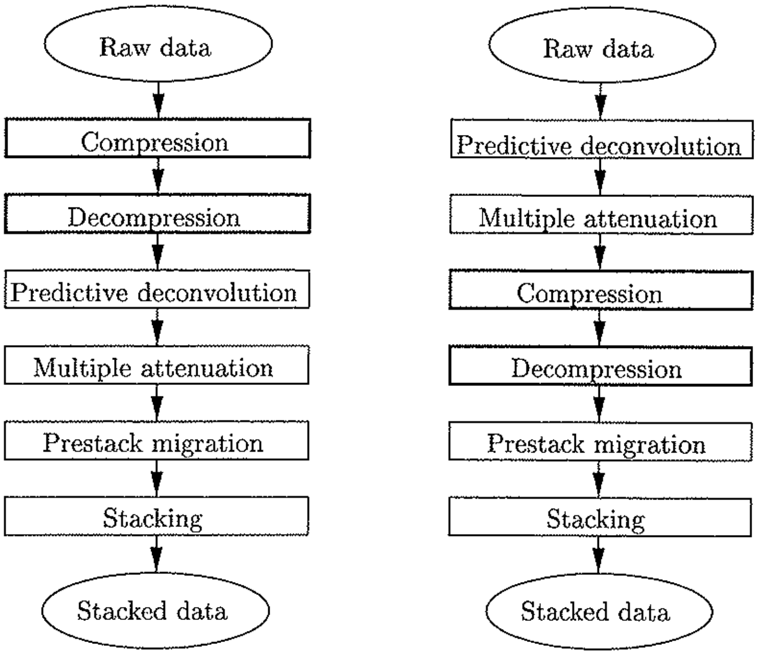
In this work, we investigate the use of lossy seismic data compression and investigate some of its influence on processing and interpretation. The applicability of the results discussed in this study is restricted to marine seismic data; compression of seismic data acquired on land is usually more involved due to overwhelming amounts of non-white noise such as ground roll (Ergas et al., 1996). Furthermore, we restrict the theory to 2-D seismic data compression with examples of common offset gathers. We use a lossy SBC method (Røsten et al., 1999b) for the compression of common offset gathers, and a lossless compression technique based on arithmetic coding (Røsten et al., 1999c) to compress the associated trace identification headers (containing for example acquisition parameters, geometry, etc.).

We explore the following two sequences, called pre-decon and post-decon compression, to better discriminate the influence of highly data adaptive or dependent processing steps (for example predictive deconvolution and attenuation of multiple reflections) on the seismic data. The compression method itself is fixed for both sequences, the only difference between them (except from two different types of *preprocessing* schemes as explained later) is where in the processing sequence the compression and decompression algorithms are applied (see Figure 4.1):

- **pre-decon compression** – each and every of individual processing tasks, including predictive deconvolution and multiple attenuation, is performed after compression and decompression of seismic raw data;

- **post-decon compression** – possible harmful processing steps when applied to decompressed seismic data (i.e., predictive deconvolution and multiple attenuation) are carried out before seismic data compression. Processing tasks which attenuate white noise (i.e., prestack migration and stacking) are, on the other hand, performed after decompression of the seismic data.

In the following sections, we describe the seismic data set and the entire processing flow, for which the influence of compression distortion is investigated. Then we summarize the compression methods applied to the trace data samples of the seismic data and the associated trace identification headers. Compression results are given for both pre-decon and post-decon compression. In the amplitude analysis section we compare the two different sequences of predecon and post-decon compression, considering prestack (amplitude variation with offset analysis, angle stacks, etc.) and poststack amplitudes. Poststack inversion examples are given next. Finally, discussion and conclusions end the paper.



(a) Pre-decon compression.

(b) Post-decon compression.

Figure 4.1: The main part of the two processing sequences; pre-decon and post-decon compression. The compression and decompression stages are skipped in the case of original stacked data.

4.3 Description of the seismic data set

The seismic data set we consider is a 2-D surface line acquired offshore central Norway during 1992. The source was a conventional air-gun array with a total volume of 1580 cubic inches ($\approx$ 25900 cm^3). The shot interval was 18.75 m, the source depth was 5 m, and the total number of shots in the data set was 1101. Furthermore, the streamer had 240 receiver groups with a group distance of 12.5 m, and the streamer depth was 8 m. The distance between the center of the source array and the first group (near offset) was 108.5 m. Finally, the record length (two-way traveltime) was 4000 ms with sample interval 2 ms. Hence, the number of samples per trace in the data set is 2001.

As preprocessing, we conveniently decimate the data set by a factor of 2 in time applying anti-alias filtering, increasing the time sample interval from 2 to 4 ms and reducing the number of samples per trace from 2001 to 1001.

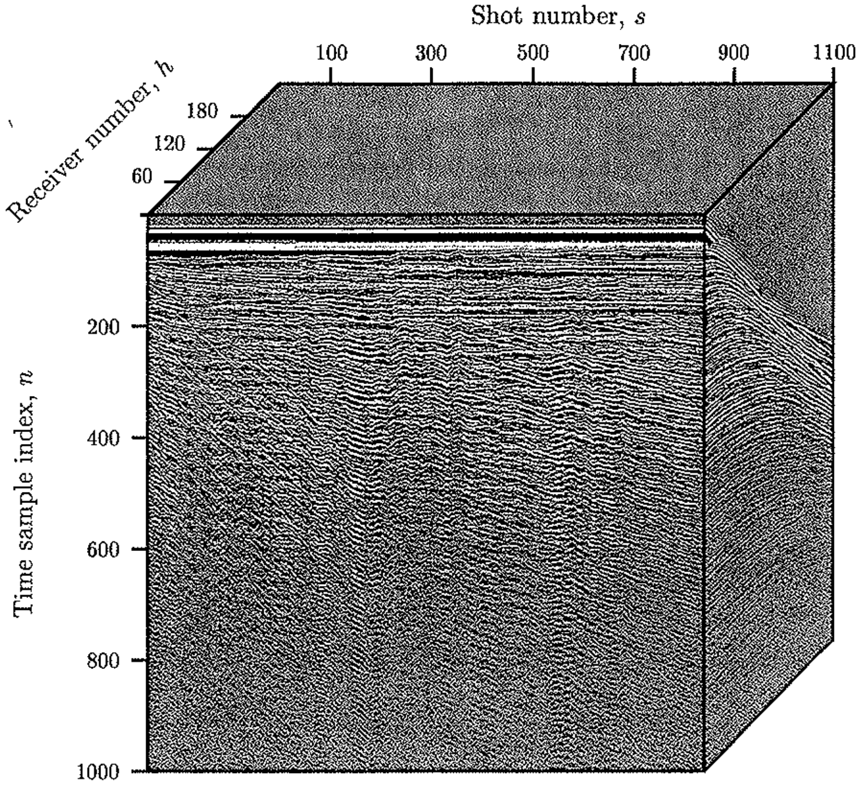


Figure 4.2: Cube of the original data set consisting of 1101 shots, 1001 samples per trace, and 240 receivers each. The first time sample index, and the first receiver and shot numbers start at 1.

At this point, we have the original representation of the seismic data. The seismic data can be viewed as a rectangular volume with axes of time, receiver (or offset), and shot. Figure 4.2 shows a cube of the original seismic data set containing 240 common offset gathers along the offset coordinate or receiver number direction.

4.4 The complete processing flow

In this study, we employ a simple, but representative processing sequence. The processing flow is carried out on original and decompressed seismic data, and the goal of the experiments is to investigate how the compression noise interferes with the seismic data during processing. The processing sequence is optimized for subsequent prestack and poststack amplitude analysis, with particular attention paid to preserving amplitude. Important steps include noise suppression, amplitude corrections, predictive deconvolution, attenuation of multiple reflections, and migration.

The processing parameters are similar for original and decompressed seismic data, but the parameters are tuned for the original data set. The stacking velocity field, which we denote the root-mean-square (rms) velocity field v_{rms}, is picked for every hundred CMP gather (or for every 625 m) based on the original data set.

The entire processing scheme can be divided into four main elements, given in sequential order:

1. processing of common shot gathers (CSGs);

2. sorting into and processing of common mid-point (CMP) gathers;

3. sorting into and processing of common offset gathers (COGs);

4. finally, sorting into CMP gathers with stacking.

4.4.1 Processing of common shot gathers

We attenuate the water-bottom multiples using a technique based on the acoustic wave equation, i.e., the so-called wave-equation-multiple-rejection (WEMR; Wiggins, 1988) method. With this data adaptive approach, the sea-floor multiples are predicted by extrapolation of the seismic data, and then the predicted multiples are adaptively subtracted from the original data. True amplitude recovery (TAR) is then applied to balance the amplitudes of the seismic data caused by geometrical spreading and absorption.

Predictive deconvolution can be regarded as a standard processing method employed to improve the temporal resolution of seismic data by compacting the seismic wavelet and attenuating ringing energy (Yilmaz, 1987). In order to give an acceptable result, conventional predictive deconvolution requires a seismic wavelet with minimum phase. Furthermore, predictive deconvolution assumes noise free seismic data; noise components have harmful effect on deconvolution (see e.g., Duarte, 1992). Hence, predictive deconvolution is – probably – the most difficult processing task to apply on decompressed seismic data, at least for high compression ratios. Only testing will determine whether deconvolution performs satisfactorily or not on seismic data with a noise problem, i.e., on seismic data with compression noise.

Summary of the processing flow ("1."):

1-a. elimination of sea-floor multiples using WEMR with a constant P-wave velocity in water of 1500 m/s;

1-b. top mute to remove direct waves and P-wave refractions;

1-c. spherical spreading correction using $v_{\mathrm{rms}}^2 \times t$, where t denotes time, and correction of absorption by 3.5 dB/s;

1-d. minimum-phase conversion of the seismic wavelet using a filter derived from the far-field signature;

1-e. predictive deconvolution using 32 ms prediction lag, 400 ms operator length, and 0.1 % additive white Gaussian noise; one design window — near offset: 700-3500 ms, far offset: 1700-4000 ms;

1-f. zero-phase conversion of seismic data after predictive deconvolution by a filter derived from the average seismic wavelet.

4.4.2 Processing of common mid-point gathers

In the CMP domain, the CMP spacing is 6.25 m, the trace spacing within an individual CMP is 37.5 m, and the maximum fold is $F = 80$.

The parabolic Radon transform is utilized on normal move-out (NMO) corrected CMP gathers for multiple attenuation (Hampson, 1986; Foster and Mosher, 1992). A parabolic form is utilized since multiples have an approximately parabolic trend after NMO. After CMP sorting of the data set, we combine three and three adjacent CMP gathers to obtain super CMP gathers, representing the full range of offsets, to avoid spatial aliasing in the parabolic Radon transform (Hugonnet and Canadas, 1995). This means that we merge three neighboring CMP gathers, each having 80 traces, into one super CMP gather containing 240 traces. Hence, the trace spacing decreases from 37.5 to 12.5 m.

The Radon demultiple method consists of forward and inverse Radon transforms in addition to top-mute operations. The multiples are estimated in the

Radon or time-moveout space and subtracted from the Radon transformed input data. Primaries ideally map from offset space to zero residual move-out in the Radon domain, while multiples map to the positive residual move-out zone.

Summary of the processing flow ("2."):

2-a. NMO correction using v_{rms};

2-b. top mute to remove dominant NMO stretch;

2-c. multiple attenuation by parabolic Radon transform using 300 parabolas sampled uniformly from -200 to 1500 ms at reference offset equal to 3096.0 m;

2-d. inverse NMO correction;

2-e. zero-phase bandpass filtering in three time gates to remove low-frequency and high-frequency noise components:[4.1] 1-4-90-110 Hz in time gate 0-1500 ms, 1-4-90-100 Hz in time gate 2500-3000 ms, and 1-4-80-90 in time gate 3500-4000 ms.

4.4.3 Processing of common offset gathers

The final processing element is migration of COGs. Migration moves dipping reflectors into their true subsurface positions and collapses diffractions, thereby increasing spatial resolution. Migration further reduces the amount of noise, including compression distortion, in the data set. Remark, *prestack* migration is utilized due to prestack amplitude analysis (Mosher et al., 1996).

The data set is binned into super common offset sections, i.e., we merge three and three neighboring COGs into super common offset sections, each containing all CMPs. Thus, the CMP spacing within the common offset sections decreases from 18.75 to 6.25 m. The super common offset panels are migrated separately.

Summary of the processing flow ("3."):

3-a. prestack Kirchhoff time migration using smoothed v_{rms} field. The maximum frequency migrated is 70 Hz, and the migration aperture is set to 3000 m;

3-b. top mute to remove seismic data migrated above the water bottom.

[4.1] The frequencies are denoted f_{s_1}-f_{p_1}-f_{p_2}-f_{s_2}, where f_{s_1} and f_{s_2} are stopband frequencies and f_{p_1}-f_{p_2} are passband frequencies.

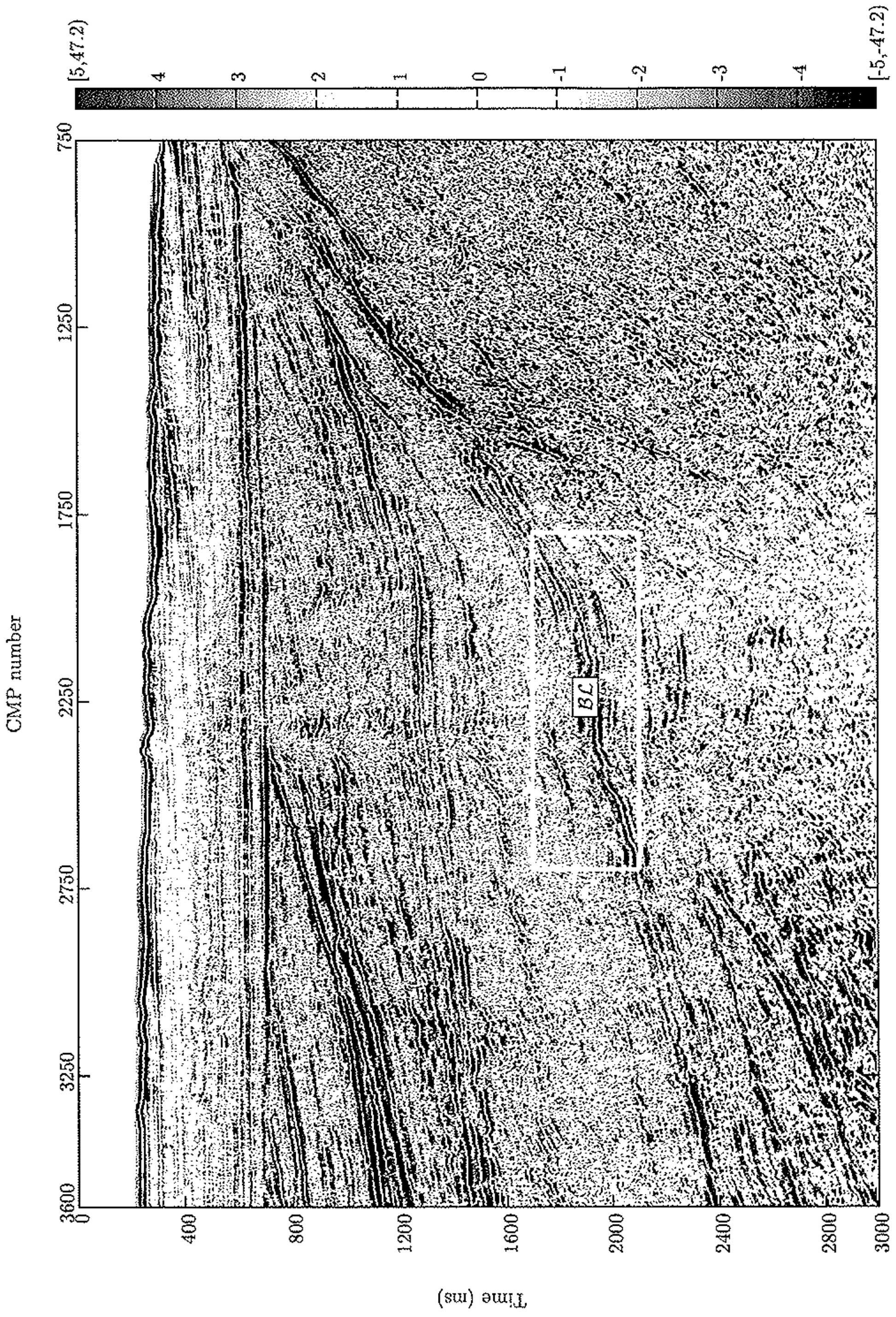


Figure 4.3: Stack section of the original data set.

4.4.4 Stacking of common mid-point gathers

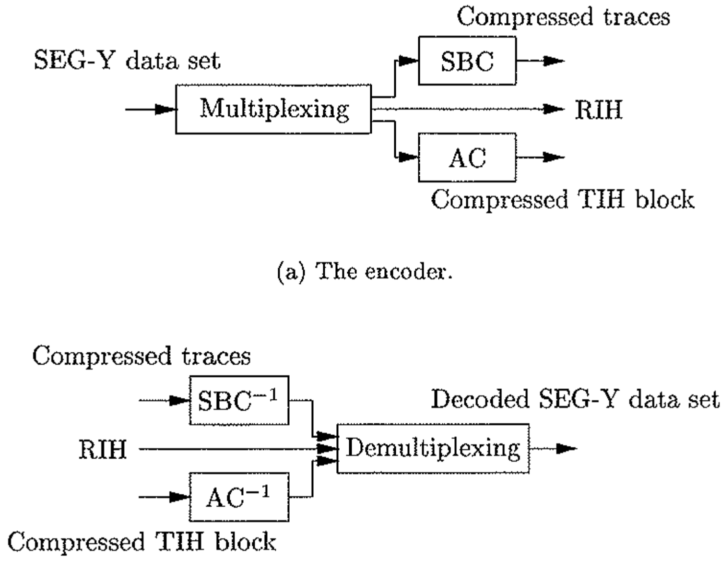
The processed data are sorted into CMP gathers which are stacked. Figure 4.3 displays the seismic image, for CMP locations from 750 to 3600 and time window from 0 to 3000 ms, after a complete processing of the original seismic data set. The stack section is of good quality. We choose "Base Lysing", denoted $\mathcal{BL}$, as a reference horizon for subsequently prestack and poststack amplitude analysis. $\mathcal{BL}$ is one of the strongest reflectors and is located in the zone between approximately 1700 and 2100 ms for CMP locations from 1800 to 2700 (see the yellow box in Figure 4.3).

4.5 Compression methods

In this section, we summarize the compression methods utilized for the seismic data (or trace data samples) and the associated trace identification headers (TIHs). For details we refer to Røsten et al. (1999b,c), respectively.

The data set, with headers, is gathered into 240 common offset sections each having 1101 number of traces and 1001 samples per trace. The original data set is represented with 32 bits per sample, i.e., 4004 bytes per trace. The size of the TIHs is 240 bytes per trace which is about 6 % of the size of the trace data samples. The total volume of the data set, including reel and trace identification headers (SEG-Y standard; SEG Tech. Std. Comm., 1997), exceeds 1.12 gigabytes. Consequently, for the data set, if the TIHs is maximally compressed $\tilde{R}:1$, then compression ratios $R:1 > 94/6 \times \tilde{R}:1$ for the trace data samples provide less marginal coding efficiency. Eventually uncompressed trace identification headers start to dominate (or become the largest part of) the data set when the trace data samples are compressed more than approximately 16:1.

At the compression or encoder side [see Figure 4.4 (a)], each of the 240 common offset gather is multiplexed into a trace data sample block (with 4004×1101 number of bytes), a reel identification header (RIH) of 3600 bytes, and a TIH block (with 240×1101 number of bytes). The trace data sample blocks and the connected TIH parts are separately compressed using a 2-D (lossy) SBC method (Røsten et al., 1999b) and a 1-D (lossless) arithmetic coding (AC) method (Røsten et al., 1999c), respectively. The RIH however is left uncompressed. At the decompression or decoder side [see Figure 4.4 (b)], inverse operations are performed in reverse order: the trace data samples and the TIHs are separately reconstructed by SBC^{-1} (using superscript -1 although not a perfect inverse operation) and AC^{-1}, respectively, and combined with



(a) The encoder.

(b) The decoder.

Figure 4.4: The complete compression algorithm.

the RIH into integral COGs. Finally, the 240 reconstructed COGs are merged into a new version of the data set, the only difference to the original data set is the compression noise introduced into the trace data samples.

4.5.1 Lossy compression of the trace data samples

The principles of 1-D SBC (Ramstad et al., 1995), at the encoder, is based on the decomposition of the input signal into M spectral subbands by an analysis filter bank to reduce the redundancy. Ideally, the analysis filter bank decorrelates the input signal, and concentrates the energy of the input signal into a minimum number of subband channels. After subband decomposition, the subband samples are down-sampled by M (every Mth subband sample is retained) to keep the total number of subband samples unchanged compared to the original signal representation. The down-sampled subband samples are quantized to eliminate the irrelevancy. The quantization step gives an approximation of the subband samples, and solely introduces the compression noise. Finally, the quantized subband samples are entropy coded (thus compression is achieved). At the decoder, the M subband samples are entropy decoded

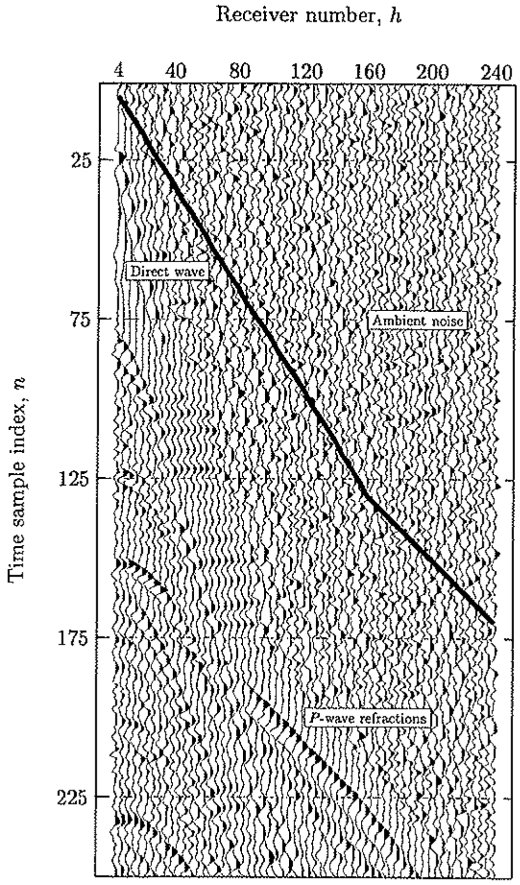


Figure 4.5: Shot number 1 for the original data set; the model of the TWT delay is given by the solid line. The display is conveniently scaled with a 100 ms standard AGC and down-sampled by 4 in the offset direction for display purposes.

and dequantized. Then the subband samples are up-sampled by M (that is inserting $M - 1$ zeros between every subband sample), and the signal is reconstructed by a synthesis filter bank at the end. 2-D SBC is implemented by separable analysis and synthesis filter banks.

To maximize the compression performance, we have to carry out *preprocessing* of the seismic data (Røsten et al., 1999b). At the encoder, we first compress the main part of the two-way traveltime (TWT) delay of the *water layer* for every COG using run-length coding (RLC; Nelson and Gailly, 1996). The run of small amplitude values in the TWT delay of the water layer (e.g., ambient noise) is set to zero and uniquely expressed by a count C which represents the number of zeros in the run. The TWT delay of the water layer is for simplicity modeled as the area in the COGs from and above a horizontal line placed at time sample index, $\tilde{n}$. For our data set, we employ this heuristic and continuous model which is found by visual inspection:

$$\tilde{n} = \begin{cases} \lceil 0.8 \times h + 0.95 \rceil, & 1 \leq h \leq 159 \\ \lceil 0.5 \times h + 50.0 \rceil, & 160 \leq h \leq 240, \end{cases}$$

where h denotes the receiver number. The count of zeros in the run is thus given by $\mathcal{C} = 1101 \times \tilde{n}$ since every COG has 1101 number of shots. Figure 4.5 shows the model of TWT delay as a function of offset for shot number 1. As can be seen, this simple model is placed in a secure distance from, e.g., the direct wave (at small offsets) and the P-wave refractions (at large offsets). Thus, the appearance of edge effects after decompression is prevented to propagate into important areas of the seismic data. Second, in the case of pre-decon compression, we apply a 2-D automatic gain control (AGC) to diminish the "non-stationary" behavior of the COGs. The AGC is efficiently implemented by calculating the mean of absolute values of $B \times B = 32 \times 32$ adjacent non-overlapping signal samples. These AGC values are reshaped to a vector and coded using a 1-D fixed-rate open-loop differential pulse code modulation (DPCM; Gersho and Gray, 1991). Piecewise-linear interpolation between the *decoded* mean of absolute values is used separately in the temporal and spatial directions to give a complete AGC function. In the case of post-decon compression, on the other hand, AGC is not applied since the seismic data set is already balanced by TAR.

After preprocessing, the common offset gathers are compressed using a 2-D SBC method (Røsten et al., 1999b). We use $M \times M = 8 \times 8$ separable parallel-structured uniform finite impulse response (FIR) analysis and synthesis filter banks. The decomposition is fixed, i.e., parallel and uniform, but the coefficients of the filter banks are adapted to the temporal and spatial correlation of COGs. The filter banks are optimized to maximize the so-called coding gain, while securing near perfect reconstruction in the absence of quantization noise (Aase, 1993). At the encoder, every COG is decomposed by the separable analysis filter bank. The 64 subbands are segmented into square blocks of $\tilde{B} \times \tilde{B} = 4 \times 4$ samples from the same subband, and the variances of individual blocks are estimated. Each block is then classified into one out of $C = 5$ classes according to its variance estimate (Lervik, 1996). All subband samples are quantized in one common uniform scalar quantizer using a constant step-size. Finally, the classes of quantized subband samples are coded using $C = 5$ different variable-rate arithmetic encoders. At the decoder, the quantized subband samples are decoded by appropriate arithmetic decoders. Subsequently, the subband samples are approximated by an inverse uniform scalar quantizer. The recreated COGs are obtained from the 64 dequantized subbands by the separable synthesis filter bank, followed by inverse AGC in the case of pre-decon compression. The AGC values are achieved by an inverse fixed-rate open-loop DPCM, and the complete AGC function is found by piecewise-linear interpolation applied separately in the temporal and spatial directions. The TWT delay of zero amplitude values is finally put back on every COG, totally removing the original inherent ambient noise in this time interval of the seismic data.

4.5.2 Lossless compression of the trace identification headers

The associated trace identification headers are compressed using a 1-D lossless AC method (Røsten et al., 1999c). First, at the encoder, the TIH block is scanned byte-wise across the traces within every COG, starting with byte number 1 and ending with byte number 240 to organize the TIHs in a natural manner for compression. Second, 1-D first-order fixed differential coding (DC) is utilized to decrease the redundancy between the bytes after the scanning procedure. Third, we utilize RLC to take advantage of bytes from the DC being constants across many traces. As the core of the compression method, we finally use *conditional adaptive* multi-alphabet AC of order $O = 8$. At the decoder, we perform the inverse operations in reverse order: inverse conditional adaptive AC, inverse RLC, and 1-D first-order fixed inverse DC followed by a rescanning procedure at the end.

4.6 Results after compression and decompression

We compress the seismic data set at four compression ratios, 7.5:1, 15:1, 30:1, and 45:1, spanning the range of practical compression ratios from low to high. The compression ratio (CR) is simply defined as the size of the data set, including reel and trace identification headers, before compression, divided by the size of the data set after compression. Since we employ variable-rate arithmetic coding, the achieved compression ratio is highly data dependent for the same quantizer settings. The step-size of the uniform scalar quantizer has to be tuned iteratively until the desired *overall* CR, including $R : 1$ and $\tilde{R} : 1$ for all COGs, is reached; decreasing the quantizer step-size will lower $R : 1$ while $\tilde{R} : 1$ is not tunable. Thus, an exact total CR is difficult to accomplish. The four desired compression ratios are therefore approximative (or average), and we denote them $\overline{7.5:1}$, $\overline{15:1}$, $\overline{30:1}$, and $\overline{45:1}$, respectively.

Immediately after compression and decompression, we calculate the full signal-to-quantization-noise ratio (SQNR; Vermeer, 1999) between the *trace data samples* of the original and reconstructed data sets, and use the SQNR as a quantitative measure of compression quality. We use the term SQNR to distinguish it from the more conventional expression signal-to-noise ratio (SNR). The SNR is commonly used for the purpose of quality measure of seismic data in the presence of ambient noise (see e.g., Junger, 1964).

The SQNR – in dB – is defined as:

$$\text{SQNR} = 20 \times \log_{10}\left(\frac{\text{rms}}{\text{rmse}}\right),$$

where rms is the root-mean-square value of the original trace data samples, and rmse is the root-mean-square error between the original and decompressed seismic data. The fraction between rms and rmse is doubled per 6.02 dB increment of SQNR. The scalar SQNR is an average measure of compression quality and can unfortunately not give any localized information about the degradation of the compression noise. Nevertheless, the SQNR provides a good relative indication of the preferred grading between different compression ratios or coding algorithms.

An SQNR exceeding 15-20 dB provides for all practical purposes a non-visible compression degradation.

4.6.1 Compression of the trace data samples

The amplitudes of the seismic data before compression are balanced by AGC (pre-decon compression) or TAR (post-decon compression). Therefore, we employ a constant quantizer step-size for all common offset gathers to equalize the quantization-noise energy which is important for prestack amplitude analysis. Nevertheless, $R:1$ for the trace data samples will differ somewhat between the common offset gathers due to different correlation properties, but also directly owing to the compression method. For the case of post-decon compression, the RLC of the TWT delay becomes continuously more efficient for larger offsets, giving typically increasing $R:1$ as a function of receiver number (see Figure 4.6). Unfortunately, for the case of pre-decon compression, the increasing efficiency of the RLC of the TWT delay is more or less counterbalanced by enlarged amount of ambient noise as a function of offset, mainly because no top-mute or bandpass-filtering operations have been applied. The case of pre-decon compression also has slightly higher amount of side information, and

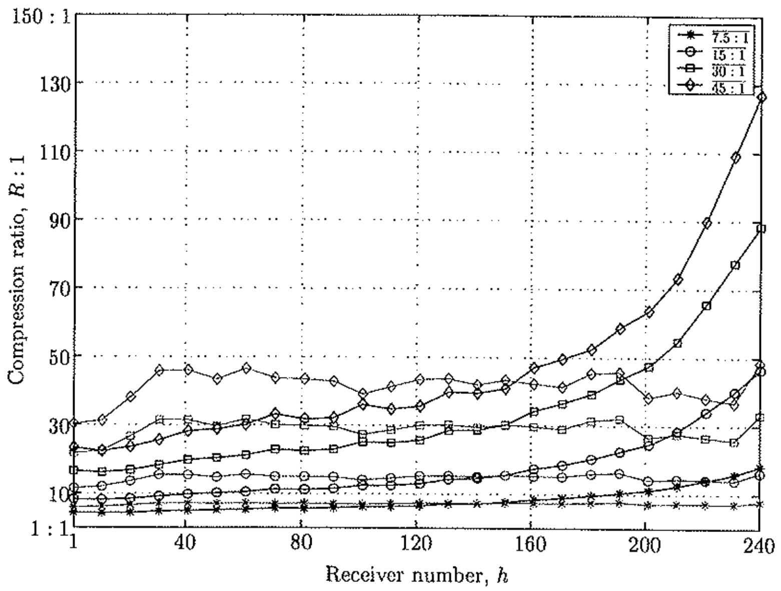


Figure 4.6: Compression ratio of the trace data samples of the common offset gathers, R : 1, versus offset at four overall compression ratios; pre-decon (red lines) and post-decon (blue lines) compression.

therefore less amount of main information for a fixed CR, due to the AGC procedure.

In general, despite the use of fixed quantizer step-size and amplitude balancing, the SQNR slightly decreases (for a given total CR) as a function of offset for both pre-decon and post-decon compression (see Figure 4.7). The rms value of the original seismic data simply drops, i.e., the size of the TWT delay of the water layer having *small amplitudes* increases, for incrementing receiver number.

The SQNR results are difficult to compare objectively, at least for high compression ratios. In pre-decon compression, for example, no gain is applied to the seismic data after decompression, (i.e., AGC-AGC^{-1}). In post-decon compression, on the other hand, the data set is balanced by TAR. A well balanced compressed-decompressed data set tends to have lower SQNR then an improperly balanced data set despite the fact that the actual or visual quality on well balanced data sets is better (Reiter, 1996). At $\overline{30:1}$ and $\overline{45:1}$, pre-decon and post-decon compression have corresponding SQNR results. However, the SQNR-versus-offset gradient is slightly steeper for post-decon compression which can be explained by the highly increasing R : 1 as a function of receiver number (see Figure 4.6). On the other hand, the compression noise

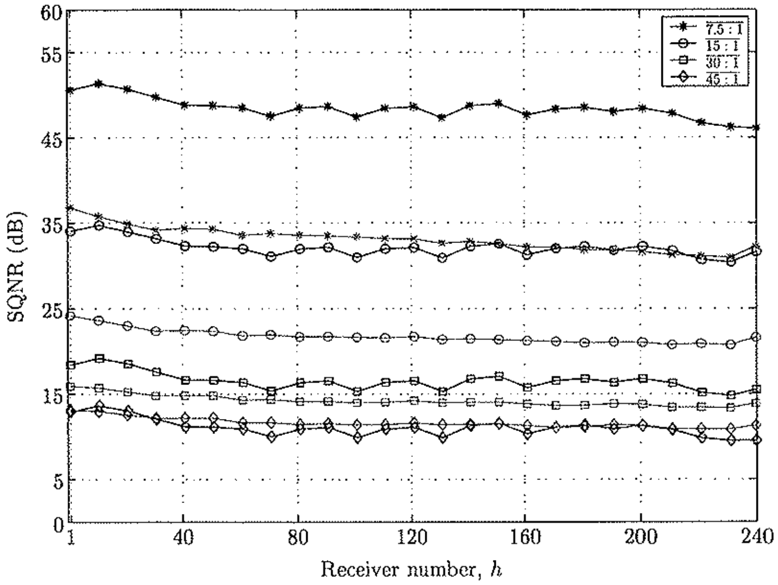


Figure 4.7: SQNR between the original and reconstructed trace data samples of the common offset gathers versus offset immediately after compression and decompression at four overall compression ratios; pre-decon (red lines) and post-decon (blue lines) compression.

is more correlated for pre-decon compression. At $\overline{7.5:1}$ and $\overline{15:1}$, the reconstructed seismic data are without visual degradation due to compression noise. Nevertheless, post-decon compression has now at least 10 dB better SQNR results due to more redundant data sets.

4.6.2 Compression of the trace identification headers

The CR of TIHs, $\tilde{R}:1$, is highly regular for both pre-decon and post-decon compression (see Figure 4.8). The utilized seismic data processing scheme before compression leads to extended amount of – and more random – trace identification header information in the latter case. In other words, pre-decon compression has the benefit of higher redundancy in the TIHs. Nevertheless, $\tilde{R}:1$ is, without exception, so high that the overall CR is more or less solely dependent upon the CR of trace data samples, $R:1$, for both pre-decon and post-decon compression.

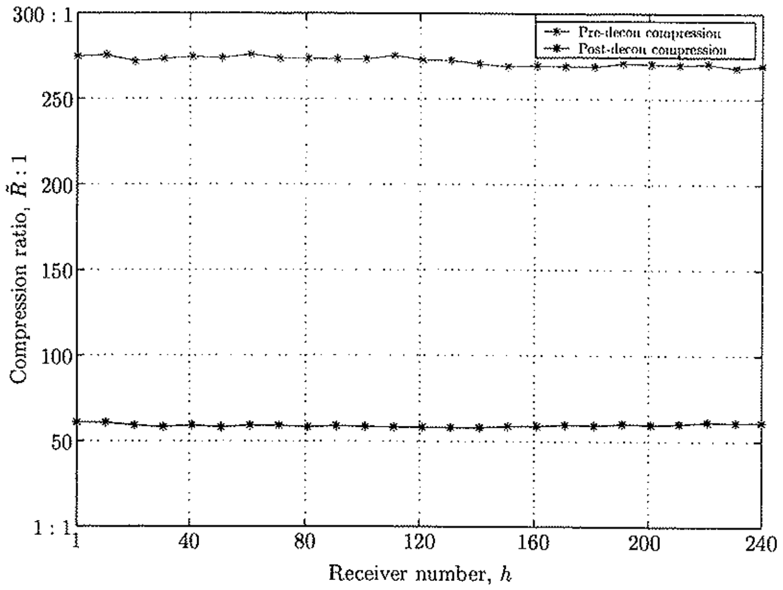


Figure 4.8: Compression ratio of the trace identification headers of the common offset gathers, $\tilde{R}$: 1, versus offset; pre-decon (red line) and post-decon (blue line) compression.

4.7 SQNR results after the processing flow

Up to now, we have examined the compression results, using quantitative SQNR measurements of the trace data samples of the common offset gathers, immediately after compression and decompression (or "before" the processing flow). The optimal information about the earth's subsurface emerges only after the seismic data have been subjected to a number of processing steps. Ultimately, our assessment of the performance of compression has to be done on seismic data after they have been processed by the complete processing flow; we proceed by comparing the compression noise immediately after compression and decompression with the compression noise that remains after the complete processing sequence.

After processing, it is most natural to study CMP gathers because this domain of seismic data is commonly used for subsequently prestack amplitude analysis and stacking. Hence, we investigate how the compression noise introduced into the COGs propagates to the CMP gathers. In practice, we divide the evaluation in accordance with the utilized processing flow, and further split the assessment into the cases of pre-decon and post-decon compression for all four compression ratios. Specifically, we evaluate the decompressed

CMP gathers before processing and after processing elements (see section 4.4) "1.", "2.", "3.", and "4." in the pre-decon compression case, while we evaluate the decompressed CMP gathers prior to migration and after migration and stacking in the post-decon compression case. As a quantitative measure, we simply calculate the mean value of the SQNR results over the range of NMO corrected CMP gathers at locations from 750 to 3600. The mean SQNR values are rounded to the nearest integers. NMO correction is not applied to already migrated CMP gathers. An identical top mute operation of corrected CMP gathers is utilized for all cases to provide the most fair comparison. Based on a similar argument, TAR is always included in the SQNR results.

We separate the discussion into three important issues: (i) attenuation of multiple reflections and predictive deconvolution (processing elements "1." and "2."), (ii) migration (processing element "3."), and (iii) stacking (processing element "4.").

4.7.1 Multiple attenuation and predictive deconvolution

The evaluation of multiple attenuation and predictive deconvolution is reasonably restricted to the case of pre-decon compression only (see Table 4.1).

4.7.1.1 WEMR and predictive deconvolution

After WEMR and predictive deconvolution, the compression noise is enlarged compared to before processing for all levels of compression (see the column marked with "1." in Table 4.1). As mentioned earlier, ambient noise and compression error components can have a harmful effect on deconvolution (Chen, 1995; Guo and Burrus, 1996; Polzer et al., 1997). WEMR is data adaptive and can be considered as an extension of conventional predictive deconvolution (Zhou and Greenhalgh, 1994), and these two operations therefore interfere similarly on the compression noise.

4.7.1.2 Radon multiple rejection

The compression noise is further increased after Radon demultiple for all levels of compression noise (see the column marked with "2." in Table 4.1). Like a frequency-wavenumber filter, Radon demultiple acts as a smoothening filter, removing coherent multiples and other noise components with high spatial frequencies (Russell et al., 1990a,b; Mosher et al., 1996). Unfortunately, the Radon transform itself is highly data dependent and performs on decompressed

Table 4.1: Mean value of the SQNR results for corrected CMP gathers in the case of pre-decon compression at four overall compression ratios.

	SQNR (dB); pre-decon compression				
	Before proc.	After proc.			Stack
		"1."	"2."	"3."	"4."
CR					
7.5 : 1	31	26	24	27	33
15 : 1	18	14	13	16	23
30 : 1	10	6	5	8	14
45 : 1	7	4	3	5	11

Table 4.2: Mean value of the SQNR results for corrected CMP gathers in the case of post-decon compression at four overall compression ratios.

	SQNR (dB); post-decon compression		
	Before proc.	After proc.	Stack
		"3."	"4."
CR			
7.5 : 1	42	53	60
15 : 1	32	36	46
30 : 1	17	19	29
45 : 1	11	13	22

seismic data in the same manner as the two other applied multiple rejection methods.

4.7.2 Migration

The mean SQNR values after migration are tabulated in the columns marked with "3." in Tables 4.1 and 4.2 for pre-decon and post-decon compression, respectively. The mean SQNR value is higher after migration than before migration for all levels of compression. In other words, the migration process reduces, as expected, the amount of compression noise. This is due to the fact that the migration is a weighted summation process and therefore the noise components become somewhat weakened. The amount of noise removed by the migration is dependent on the compression ratio: the higher the compression ratio is, the more coherent will the compression noise appear, and consequently

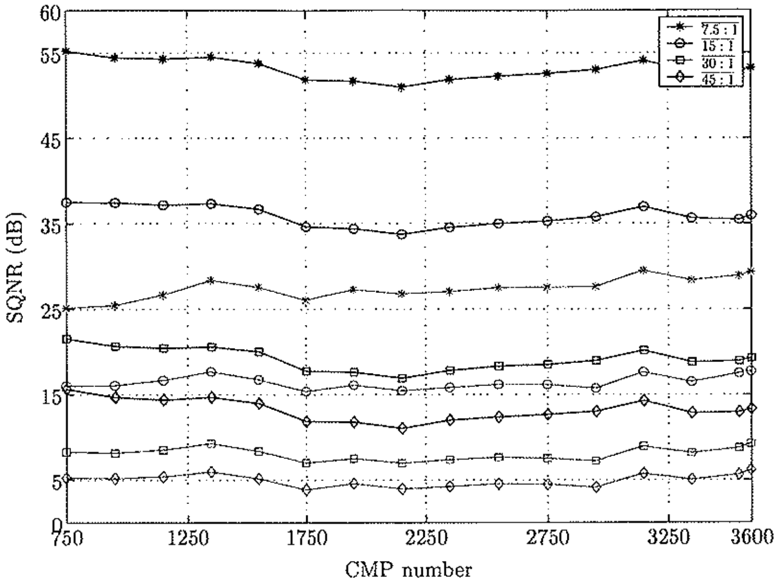


Figure 4.9: SQNR between the original and decompressed CMP gathers versus CMP number after the complete processing flow; pre-decon (red lines) and post-decon (blue lines) compression.

less compression noise will be removed by the summation process. The migration gain improvement is higher for the post-decon compression case since the compression noise is less coherent compared to the pre-decon compression case.

Finally, we plot the SQNR measurements for the migrated CMP gathers (see Figure 4.9). The SQNR is fairly flat as a function of CMP number, indicating a regular reconstruction for all CMP gathers. At $\overline{30:1}$ and $\overline{45:1}$, the SQNR results for pre-decon and post-decon compression have diverged from each other (compare with Figure 4.7) despite that identical processing flows are employed. For pre-decon compression, a compression ratio somewhere between $\overline{7.5:1}$ and $\overline{15:1}$ gives acceptable level of compression noise, while for post-decon compression, the compression noise is hardly noticeable at $\overline{7.5:1}$ and $\overline{15:1}$. Even compression ratios of $\overline{30:1}$ and $\overline{45:1}$ will provide CMP gathers with small visual appearance of compression noise in this latter case.

4.7.3 Stacking

Stacking has long been recognized as the most effective way of dealing with incoherent noise. Our stacking is done by taking averages of the sample values over corrected CMP gathers along the offset axis, using a scaling equal to the

Table 4.3: Mean value of the SQNR results for corrected CMP gathers in the case of pre-decon compression, but without WEMR, predictive deconvolution, and Radon multiple rejection.

	SQNR (dB); pre-decon compression				
	Before proc.	After proc.			Stack
		"1."	"2."	"3."	"4."
CR					
$\overline{7.5 : 1}$	31	32	33	38	44
$\overline{15 : 1}$	18	19	20	25	31
$\overline{30 : 1}$	10	10	11	15	20
$\overline{45 : 1}$	7	7	8	11	15

square-root of the number of trace data samples. Therefore, stacking is also a form of compression. The SQNR values after stacking are given in the last columns of Tables 4.1 and 4.2. As can be seen, the mean SQNR is lower before stacking than after stacking for all evaluated compression ratios. For pre-decon compression, the SQNR is increased by approximately 5 dB. For post-decon compression, however, the stacking procedure increases the SQNR in the order of 9 to 10 dB, except at $\overline{7.5 : 1}$ where the SQNR is increased by approximately 7.5 dB; this very high reconstruction quality possibly leads to a saturation in the stacking gain improvement.

4.7.4 Neither multiple rejection nor predictive deconvolution

We end this section by showing what happens to the original and decompressed seismic data in the case of pre-decon compression, if we do not incorporate WEMR, predictive deconvolution, and Radon multiple rejection in the processing flow (see Table 4.3). Most importantly, the SQNR increases (or at least does not decrease) at a cost of decreased SNR, e.g., remaining multiple reflections, for all evaluated compression ratios after all processing elements. "1." consists mainly of TAR, while "2." contains of forward and inverse NMO corrections in addition to bandpass filtering; the top mute operations remove dominant NMO stretch, but also eliminate compression noise which appears at large offsets.

4.8 Amplitude analysis

In this section we present some common seismic attributes derived from the data without any attempt to lithologic interpret or discuss the strengths and weaknesses of these attributes. The amplitude analysis is only done in the context of looking for deviations from original seismic data attributes. We assume that the acquisition and processing stages are done optimally, and apply the complete processing flow, including migration, described earlier. The reliability of the results from all kind of amplitude analysis depends on the quality of the acquisition and processing of seismic data (Russell, 1993; Buland, 1998), but also on the amount of inherent noise in the seismic data (Ross, 1993).

Table 4.4: SQNR results for conventional stack, intercept and gradient sections, and angle stacks in the case of pre-decon compression.

	SQNR (dB); pre-decon compression					
	Stack	AVO stacks		Angle stacks		
		R_0	R_1	0°-12°	12°-24°	24°-36°
CR						
7.5 : 1	33	28	26	31	32	33
15 : 1	23	19	18	20	21	22
30 : 1	14	11	10	12	13	13
45 : 1	11	8	7	8	10	10

Table 4.5: SQNR results for conventional stack, intercept and gradient sections, and angle stacks in the case of post-decon compression.

	SQNR (dB); post-decon compression					
	Stack	AVO stacks		Angle stacks		
		R_0	R_1	0°-12°	12°-24°	24°-36°
CR						
7.5 : 1	60	54	50	59	58	58
15 : 1	46	38	35	45	43	43
30 : 1	29	24	22	28	27	27
45 : 1	22	19	17	22	20	20

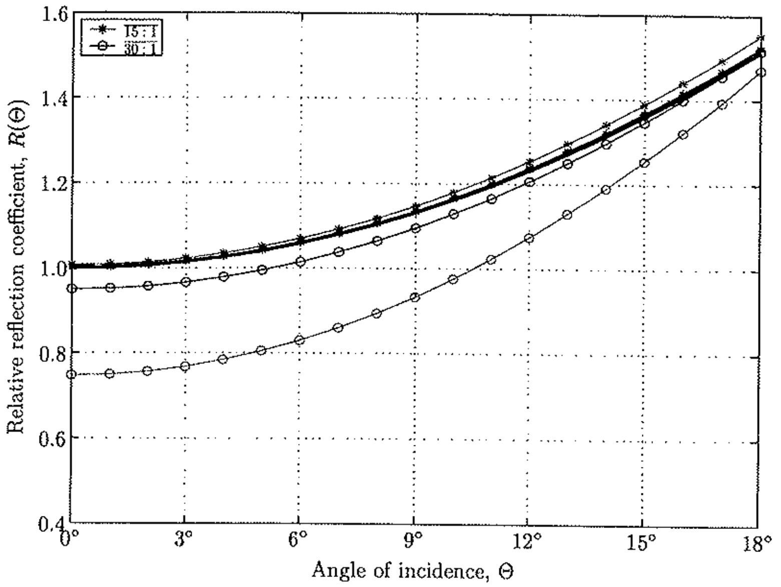
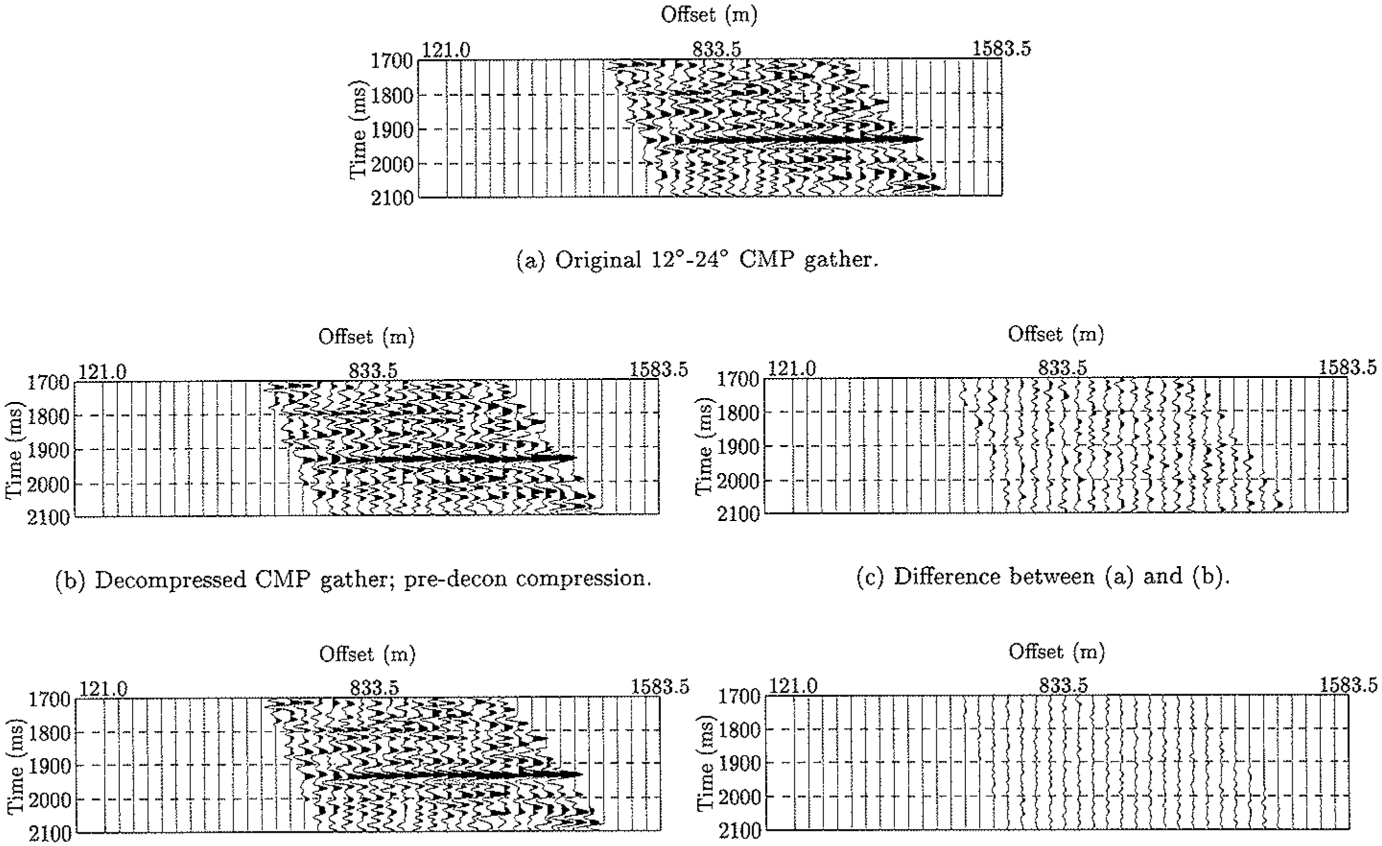


Figure 4.10: Shuey's approximation for the P-wave reflection coefficient, $R(\Theta)$, at two overall compression ratios; original data (black line), pre-decon compression (red lines), and post-decon compression (blue lines). $R(\Theta)$ is called relative and is higher than 1.0 since the intercept reflection coefficients are normalized.

4.8.1 Prestack amplitude analysis

Prestack amplitudes are used in analysis of amplitude variation with offset (AVO), where intercept, R_0, and gradient, R_1, stacks are extracted from the prestack data based on the two-term Shuey's approximation of the Zoeppritz equations for the P-wave reflection coefficient, $R(\Theta) = R_0 + R_1 \times \sin^2(\Theta)$, where Θ is the angle of incidence (Shuey, 1985; Castagna and Swan, 1997). Figure 4.10 shows the *relative* reflection coefficient as a function of incident angle, in the range from 0° to 18°, for original and decompressed data with CR of $\overline{15 : 1}$ and $\overline{30 : 1}$. The values of R_0 and R_1 are taken at the $\mathcal{BL}$ reflector for CMP location 2275. R_0 for the original data is normalized, and the intercept reflection coefficients for the decompressed data are scaled accordingly to the normalized one. A CR of $\overline{15 : 1}$ gives a well reconstructed $R(\Theta)$ for both pre-decon and post-decon compression.

The so-called angle stacks are based on partially stacking of angle limited CMP gathers; we use angles ranging from 0-12 degrees, 12-24 degrees, and 24-36 degrees. Figure 4.11 shows the original and the $\overline{30 : 1}$ decompressed 12-24 degrees angle limited CMP gathers for location 2275 in the time window from



(a) Original 12°-24° CMP gather.

(b) Decompressed CMP gather; pre-decon compression.

(c) Difference between (a) and (b).

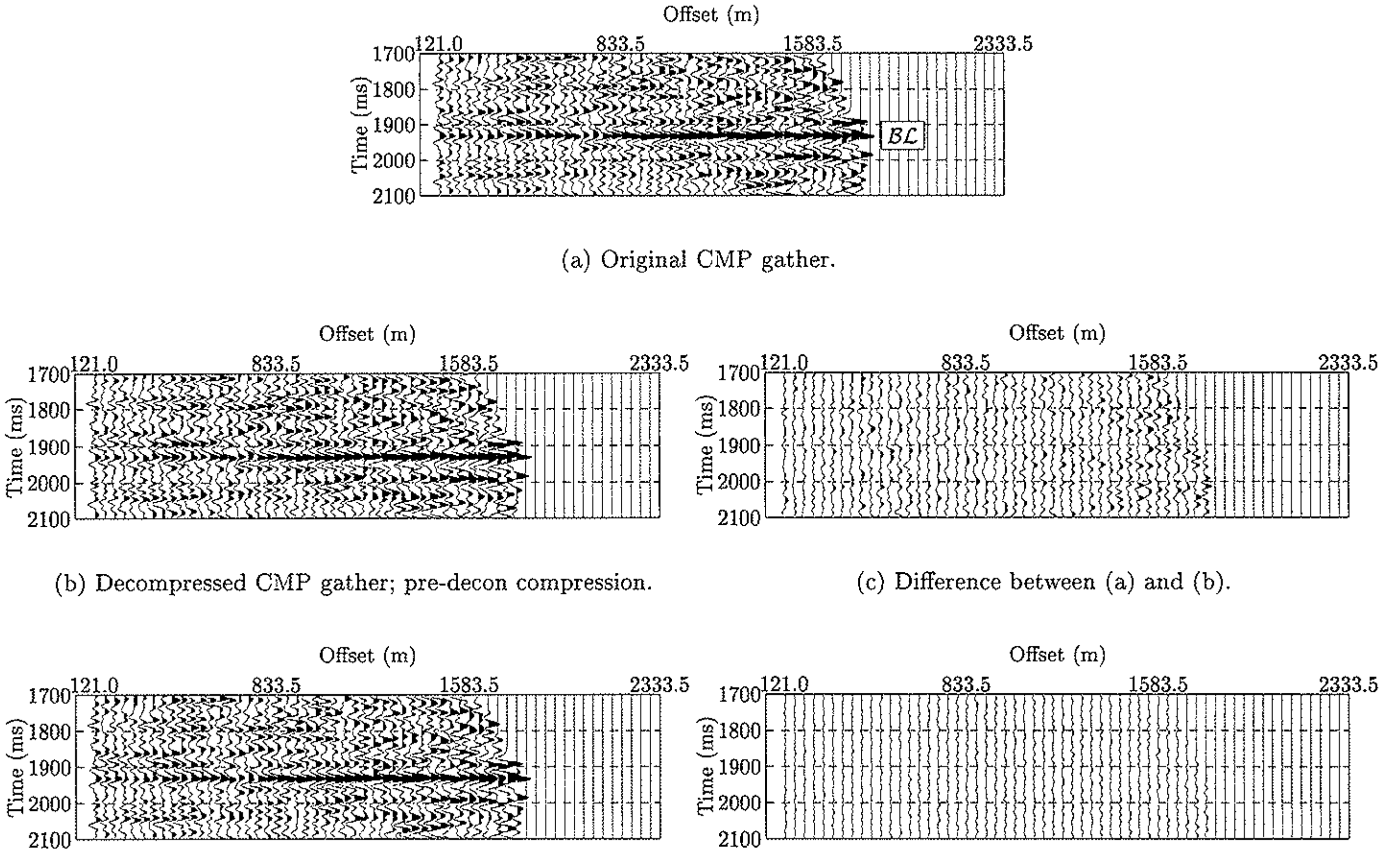
(d) Decompressed CMP gather; post-decon compression.

(e) Difference between (a) and (d).

Figure 4.11: 12°-24° angle limited CMP gathers for location 2275; CR is $\overline{30:1}$.

1700 to 2100 ms. The difference sections [see Figure 4.11 (c) and (e)] resemble small amount of, and "uncorrelated", compression noise. SQNR results for the intercept, gradient, and angle stacks are presented in Tables 4.4 and 4.5. The intercept and gradient sections have lower SQNR results than the angle stacks, but R_0 has higher SQNR than R_1. The angle stacks have more or less similar SQNR results as conventional stack sections for the case of post-decon compression, but the SQNR slightly decreases as a function of increasing angles. For the case of pre-decon compression, however, the SQNR results for the angle stacks are significantly lower than ordinary stack sections, but now the SQNR becomes larger as a function of increasing angles. It is, thus, difficult to predict universal tendencies for these fluctuating results.

More interesting than SQNR results of AVO and angle stacks are prestack amplitude maps of a single reflector. Figure 4.12 displays the original and the $\overline{30:1}$ decompressed CMP gathers for location 2275 in the time window from 1700 to 2100 ms (around the $\mathcal{BL}$ horizon). In Figures 4.13 and 4.14, the amplitude as a function of offset (i.e., the AVO response) for the $\mathcal{BL}$ reflector is displayed for pre-decon and post-decon compression, respectively; the horizon is muted above offset 1771.0 m. The reconstructed amplitudes differ significantly from the original amplitude at $\overline{45:1}$ for both instances. For pre-decon compression, the amplitude is well reconstructed at $\overline{7.5:1}$ and $\overline{15:1}$, while for post-decon compression, the reconstructed amplitude is very close (or almost identical) to the original response at $\overline{7.5:1}$ and $\overline{15:1}$and it deviates insignificantly at $\overline{30:1}$. Fortunately, the quality of the decompressed AVO responses seems to be more or less independent of offset as deduced from the applied compression algorithm [see also Figure 4.12 (c) and (e)], having a uniform scalar quantizer with a constant step-size for the different common offset gathers.



(a) Original CMP gather.

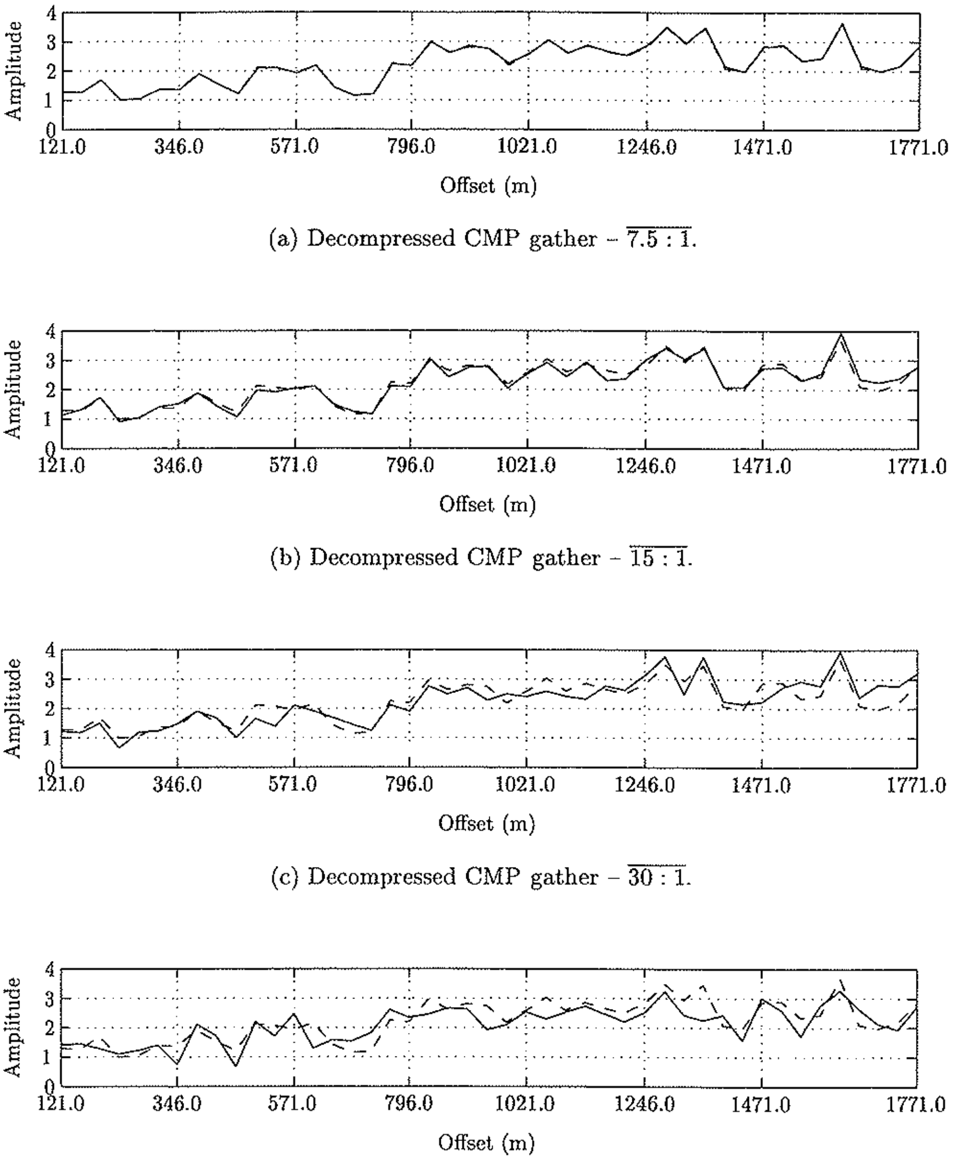
(b) Decompressed CMP gather; pre-decon compression.

(c) Difference between (a) and (b).

(d) Decompressed CMP gather; post-decon compression.

(e) Difference between (a) and (d).

Figure 4.12: CMP gathers for location 2275; CR is $\overline{30:1}$.



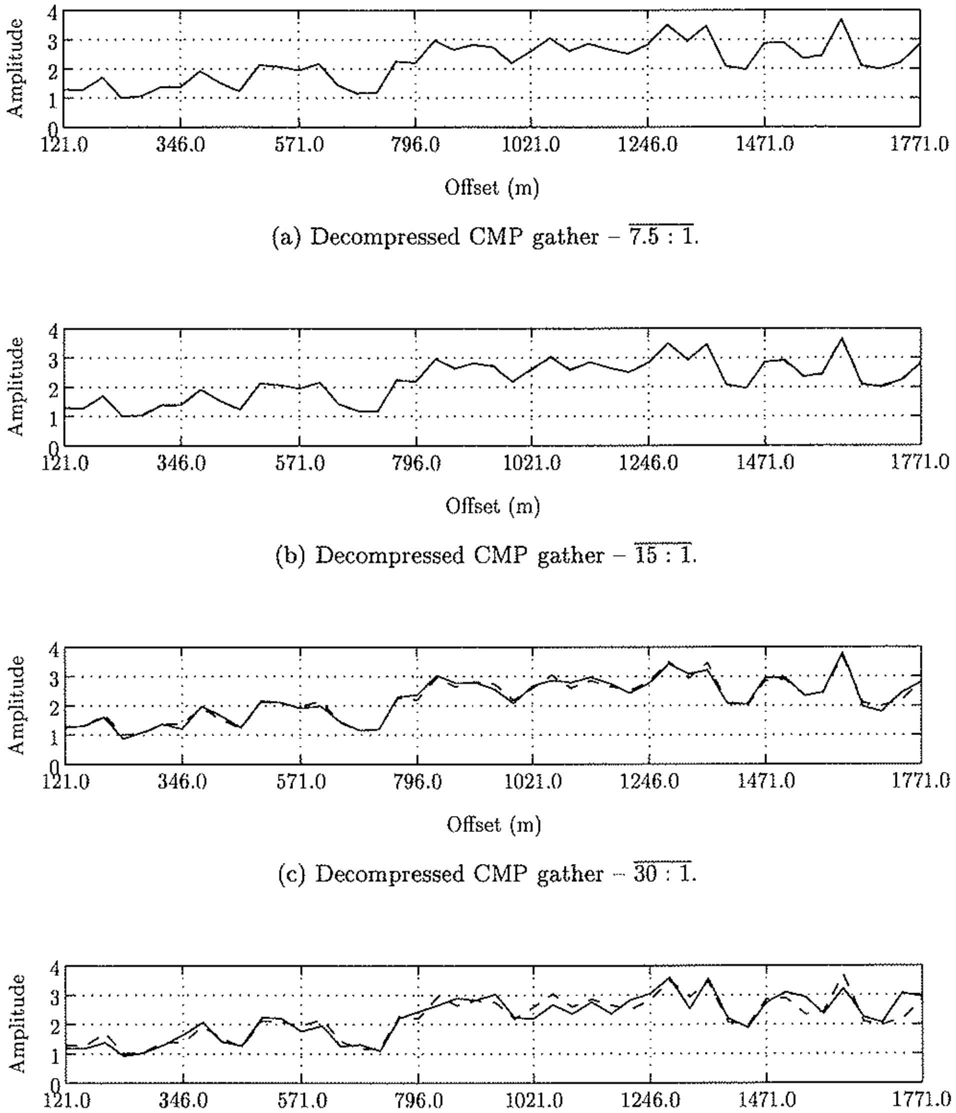
(a) Decompressed CMP gather - $\overline{7.5 : 1}$.

(b) Decompressed CMP gather - $\overline{15 : 1}$.

(c) Decompressed CMP gather - $\overline{30 : 1}$.

(d) Decompressed CMP gather - $\overline{45 : 1}$.

Figure 4.13: AVO along $\mathcal{BL}$ for CMP location 2275 in the case of pre-decon compression; decompressed (solid lines) and original data (dashed line).



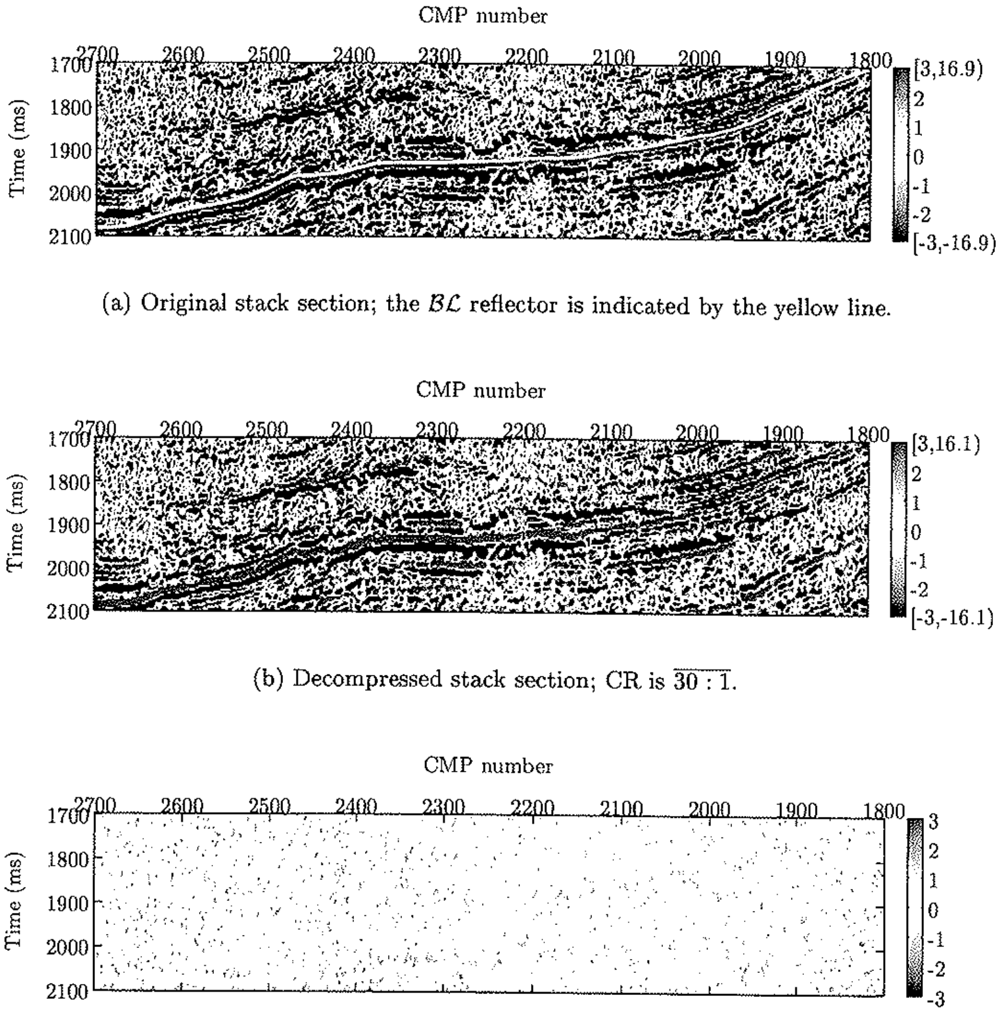
(a) Decompressed CMP gather – $\overline{7.5 : 1}$.

(b) Decompressed CMP gather – $\overline{15 : 1}$.

(c) Decompressed CMP gather – $\overline{30 : 1}$.

(d) Decompressed CMP gather – $\overline{45 : 1}$.

Figure 4.14: AVO along $\mathcal{BL}$ for CMP location 2275 in the case of post-decon compression; decompressed (solid lines) and original data (dashed line).

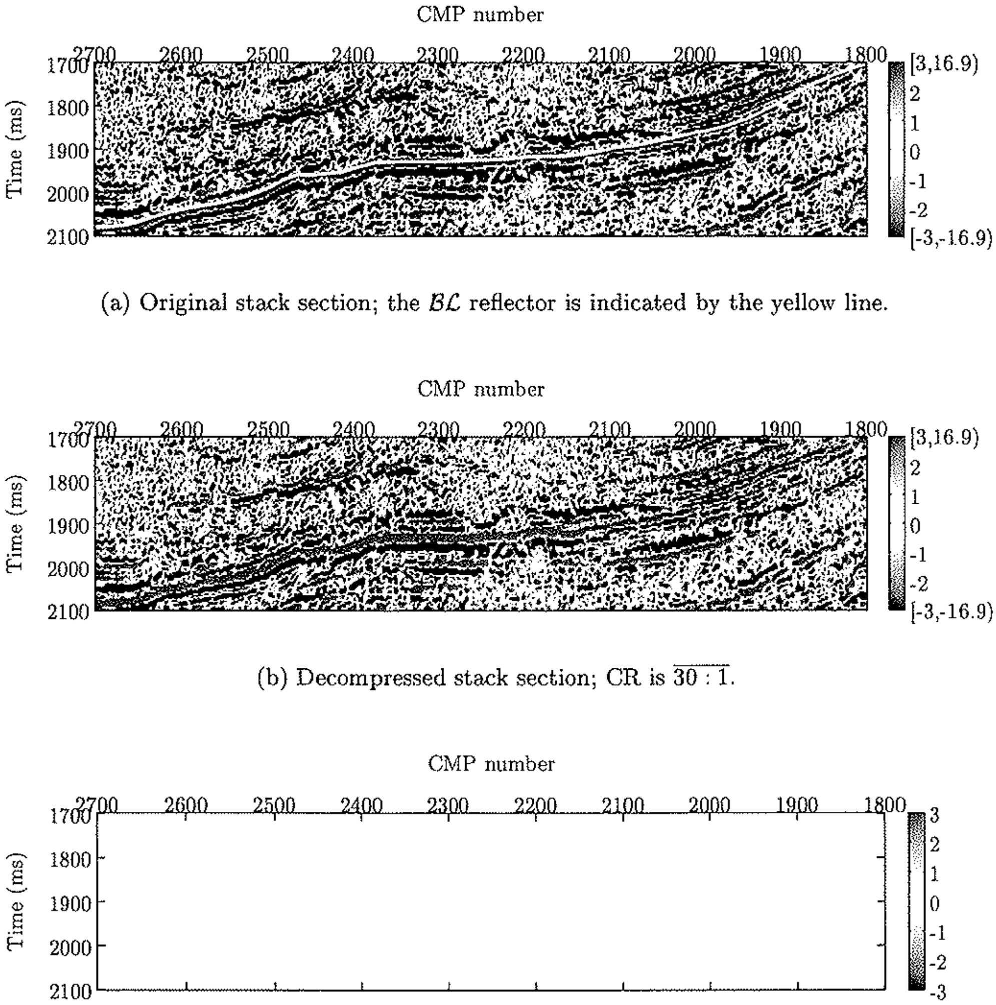


(a) Original stack section; the $\mathcal{BL}$ reflector is indicated by the yellow line.

(b) Decompressed stack section; CR is $\overline{30:1}$.

(c) Difference between (a) and (b).

Figure 4.15: Closeup views of stack sections; pre-decon compression.

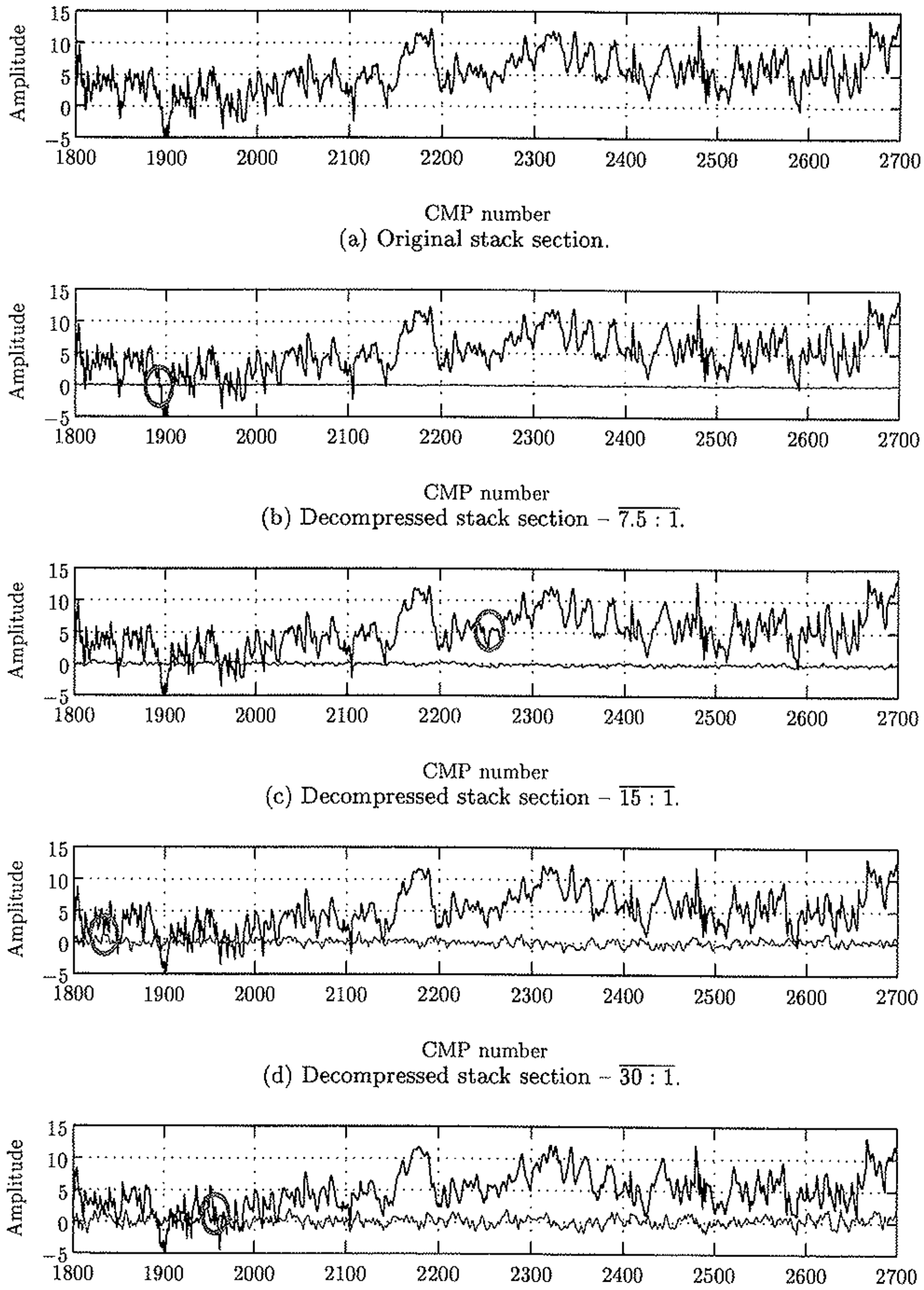


(a) Original stack section; the $\mathcal{BL}$ reflector is indicated by the yellow line.

(b) Decompressed stack section; CR is $\overline{30:1}$.

(c) Difference between (a) and (b).

Figure 4.16: Closeup views of stack sections; post-decon compression.



(a) Original stack section.

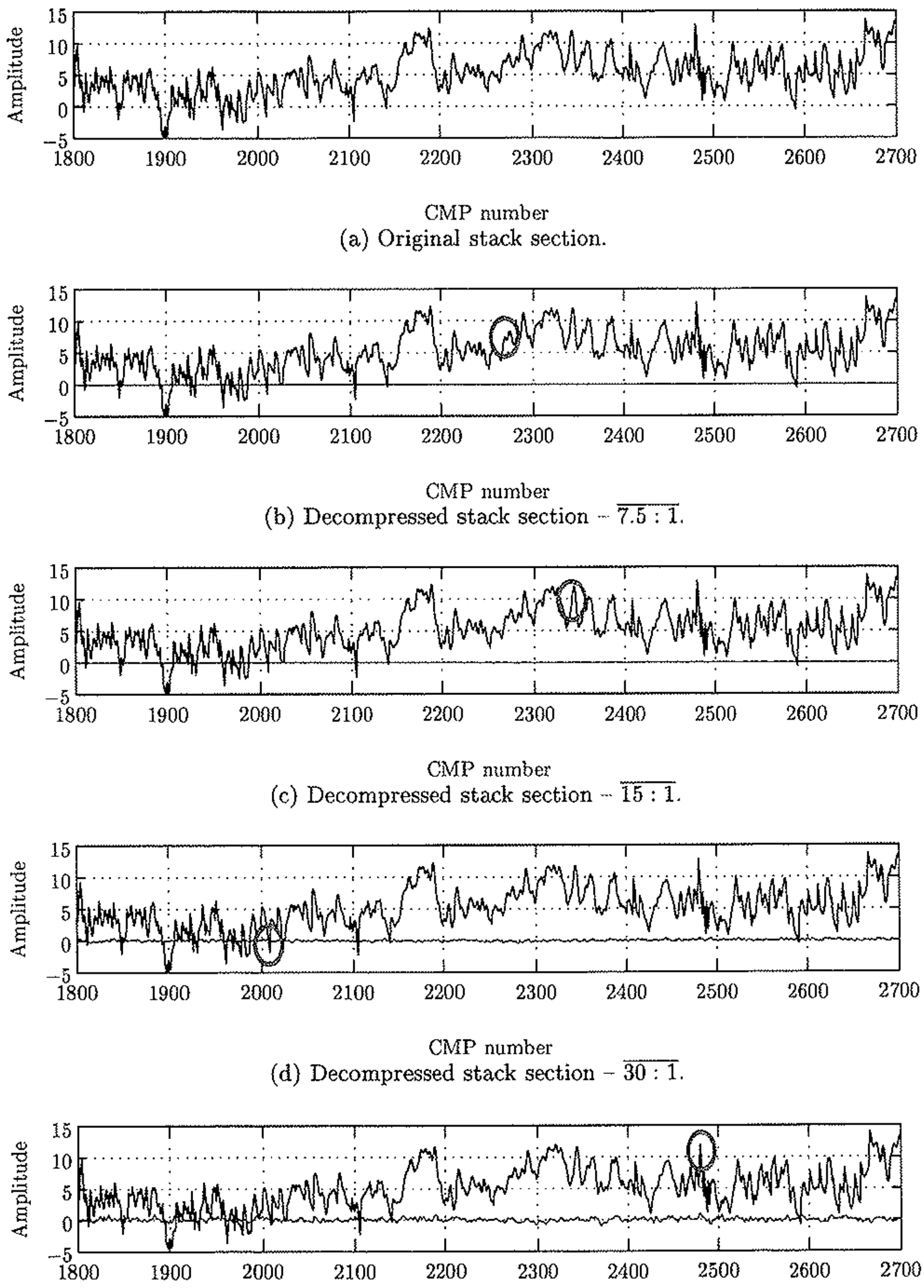
(b) Decompressed stack section – $\overline{7.5:1}$.

(c) Decompressed stack section – $\overline{15:1}$.

(d) Decompressed stack section – $\overline{30:1}$.

(e) Decompressed stack section – $\overline{45:1}$.

Figure 4.17: Stack-section amplitudes along $\mathcal{BL}$; pre-decon compression.



(a) Original stack section.

(b) Decompressed stack section – $\overline{7.5 : 1}$.

(c) Decompressed stack section – $\overline{15 : 1}$.

(d) Decompressed stack section – $\overline{30 : 1}$.

(e) Decompressed stack section – $\overline{45 : 1}$.

Figure 4.18: Stack-section amplitudes along $\mathcal{BL}$; post-decon compression.

4.8.2 Poststack amplitude analysis

For several kinds of attribute mapping and for poststack inversion, information of amplitudes in stack sections is used. Figure 4.3 displays the migrated stack image from 0 to 3000 ms. Closeup views in CMP interval from 1800 to 2700 and time window from 1700 to 2100 ms for the $\overline{30:1}$ decompressed sections are given in Figures 4.15 and 4.16. The original stack is given in panel (a) of both figures.

We analyze the stack-section amplitude along the $\mathcal{BL}$ reflector [see Figure 4.15 (a) or Figure 4.16 (a)]. The resulting amplitudes are shown in Figures 4.17 and 4.18 for pre-decon and post-decon compression, respectively. The original stack-section amplitude is given in panel (a) of both figures. The amplitude curves are highly oscillating, but all responses have similar trends and the differences between the original and the reconstructed amplitudes (marked with red lines) are small. To better discriminate the five amplitude curves from each other, the CMP locations having the largest absolute-deviation from the original amplitude response are indicated by green circles. The stacking procedure removes compression noise so well that the difference is negligible up to $\overline{15:1}$ and up to $\overline{30:1}$ in the case of pre-decon and post-decon compression, respectively.

4.9 Poststack inversion

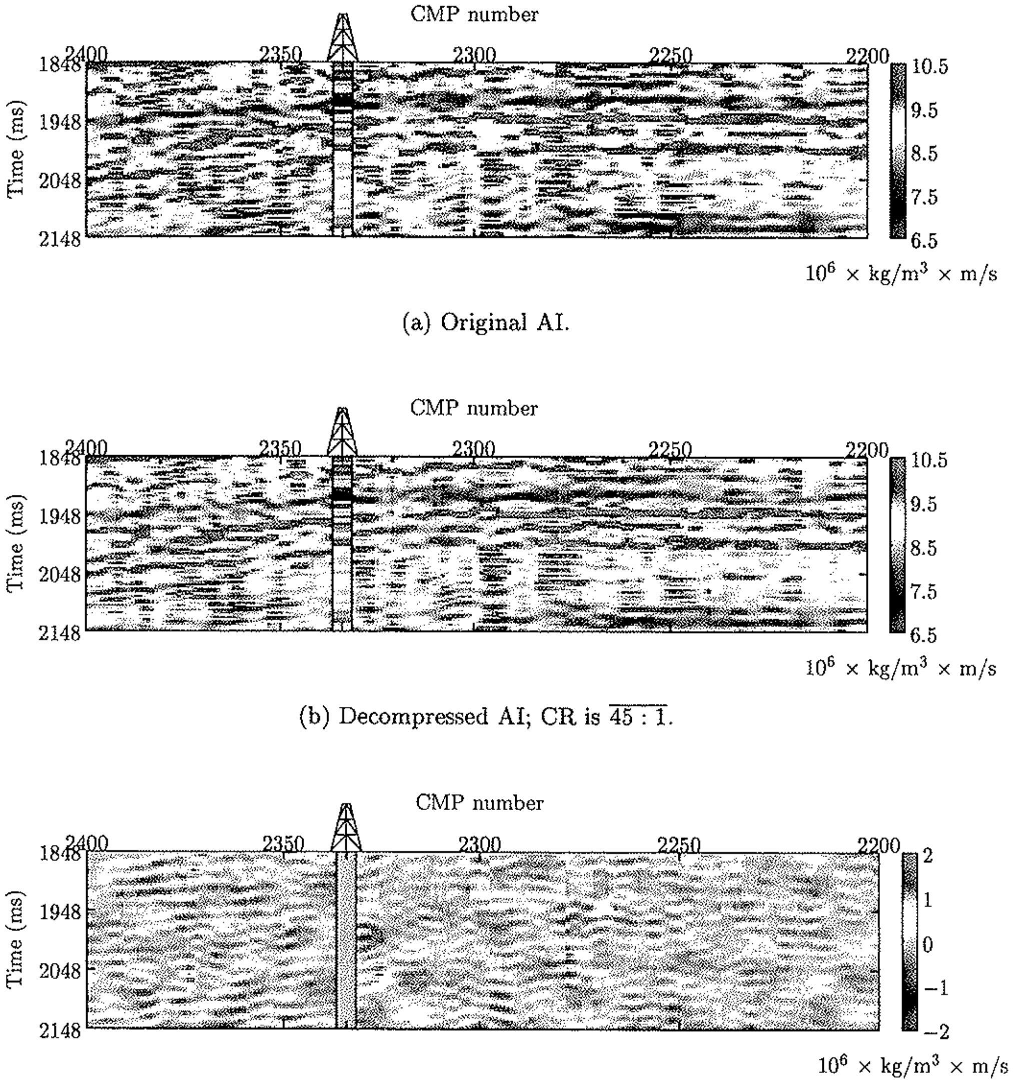
Poststack inversion provides the ability to derive estimates of acoustic impedance (AI). AI is defined as density times P-wave velocity, and among others, AI can be used to estimate porosity.

We utilize a so-called sparse spiking model-based inversion algorithm (see e.g., Latimer et al., 2000). This AI estimation method includes well log calibration against the seismic data: we create the low-frequency acoustic impedance section based on the calibrated well log using three interpreted horizons (one of which is the "Base Lysing" reflector). The relative acoustic impedance (RAI) sections, i.e., AI without the low-frequency background model, are estimated for all four compression ratios in the case of both pre-decon and post-decon compression. The optimum seismic wavelet used in the RAI estimation is calculated separately for all input full-fold stack sections. We calculate the perturbation of AI only where well log information of density and P-wave velocity is available, i.e., in the time window from 1848 to 2408 ms. The estimation of the (absolute) AI is at the end given by addition of the low-frequency acoustic impedance section to the RAI.

Absolute AI sections for original and decompressed data, in a target zone of CMP locations from 2200 to 2400 and time window from 1848 to 2148 ms, are shown in Figures 4.19 and 4.20. The CR is $\overline{45 : 1}$, and the original AI is given in panel (a) of both figures. The AI well log information is additionally inserted for quality control. Ideally, we should compare with well logs not used in the inversion, but no other well logs are available. The well log is not directly placed on the seismic line, but is located in a distance about 500 m off the seismic line. Therefore, as an approximation, the well log is projected to the nearest CMP location (2334). The match between the AI well log and the estimated AI is nevertheless satisfactorily. The main issue, however, is to compare the AI sections estimated from original and decompressed seismic data with each other. The compression noise is significant at high compression ratios [see Figure 4.19 (c) and Figure 4.20 (c)]. The SQNR results for absolute and perturbationed acoustic impedance sections are given in Table 4.6. AI has only positive values, the only difference to the zero-mean RAI is the low-frequency background model. Thus, the rms (and the SQNR) value of absolute AI is higher than for RAI, but the rmse values between original and decompressed absolute and relative acoustic impedance sections are identical. For pre-decon compression, a CR of $\overline{7.5 : 1}$ provides a good reconstruction quality of AI, while a CR of $\overline{15 : 1}$ gives acceptable AI results for the case of post-decon compression.

Table 4.6: SQNR results for relative and absolute acoustic impedance sections.

	SQNR (dB)			
	Pre-decon compression		Post-decon compression	
	RAI	AI	RAI	AI
CR				
$\overline{7.5 : 1}$	24	48	29	53
$\overline{15 : 1}$	18	42	26	50
$\overline{30 : 1}$	13	36	21	45
$\overline{45 : 1}$	8	32	14	38

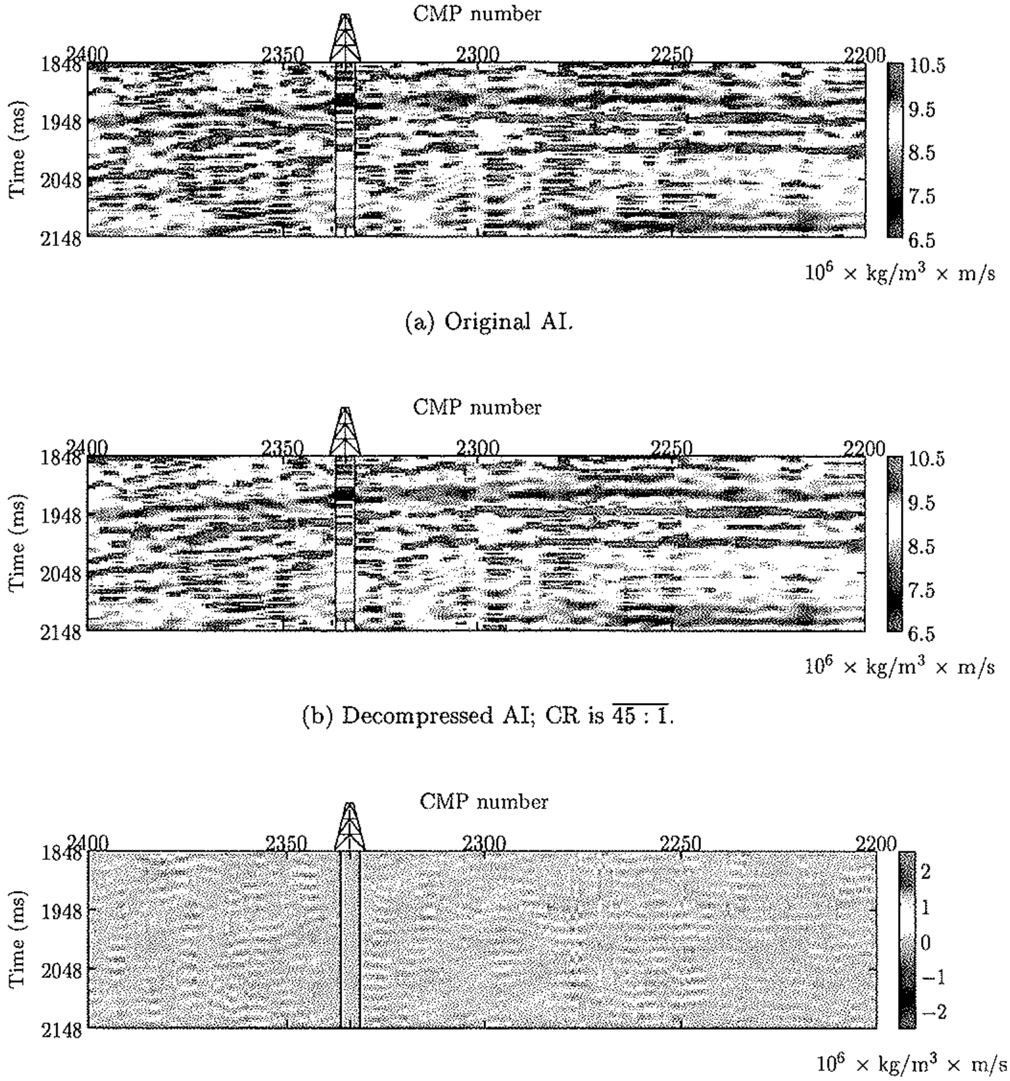


(a) Original AI.

(b) Decompressed AI; CR is $\overline{45:1}$.

(c) Difference between (a) and (b).

Figure 4.19: Acoustic impedance sections; pre-decon compression.



(a) Original AI.

(b) Decompressed AI; CR is $\overline{45:1}$.

(c) Difference between (a) and (b).

Figure 4.20: Acoustic impedance sections; post-decon compression;

4.10 Discussion

Common offset gathers are convenient to compress for several reasons. First, common offset gathers have higher redundancy and are therefore better suited for compression than common shot gathers. Second, the direct wave and/or the sea-floor reflection (at small offsets) and the P-wave refractions (at large offsets) are easily detected on COGs. Thus, the TWT of the water delay containing mostly ambient noise can be isolated and compressed separately from the rest of the seismic data. Nevertheless, a common offset gather consists of traces from all common shot gathers, a single trace at a particular offset for each common shot gather. All common shot gathers have to be acquired before they eventually can be resorted into common offset gathers. Consequently, the compression of common offset gathers is primarily applicable for the purpose of storage rather than transmission.

Specifically, we have detected the main part of the TWT delay of the water layer by simple visual inspection of the seismic data, making a heuristic model of the TWT delay. Alternatively, one can find this TWT delay by first-break-arrival detection methods (e.g., using neural networks), or simply as the traveltime of the direct wave having knowledge about the offset and using the average velocity of water (e.g., 1500 m/s). This last approach can, however, remove P-wave refractions at large offsets. In our case, the TWT delay is compressed by RLC, totally removing the original inherent information in this time range of the seismic data. The pre-first-break arrivals, however, are sometimes used to estimate the level of, and the characteristics of, the ambient noise. Hence, such an analysis of the ambient noise properties has to be performed before an eventual compression of the TWT delay by RLC.

In general, at low compression ratios, quantization noise can be seen as additive white uniform noise (Gersho and Gray, 1991). The seismic data, on the other hand, tend to be "Gaussian" distributed. Thus, the error introduced into the seismic data during compression also tends to be additive white "Gaussian" noise. The effect of additive (colored or white) noise on traditional predictive deconvolution is strongly dependent on how the power spectrum of the noise compares to that of the seismic data. A very favorable condition is when the additive noise has a power spectrum similar to that of the seismic data (Duarte, 1992). In the case of additive white Gaussian compression noise, the level of the noise's power spectrum should preferably be below the level of the power spectrum of the seismic data. This observation has been stressed by Polzer et al. (1997), who conclude that amplitude balancing and emphasis filtering can counter the effect of compression noise on deconvolution. Guo and Burrus

(1996) focus, on the other hand, on the use of seismic data compression techniques that preserve the phase information as much as possible. They argue that the seismic wavelet can be transformed from minimum-phase to mixed-phase during compression, causing wrongness in the predictive deconvolution result. Of these two solutions, the first suggestion simply needs the use of some kind of filtering routines, while the second proposal requires a more complex compression algorithm.

Chen (1995) argues that predictive deconvolution is better to utilize *after* compression and decompression, in direct opposition to our experience. Chen reasons that deconvolution broadens the power spectrum of the seismic data, making the seismic data more difficult to compress. This reduced coding efficiency, if deconvolution is applied before compression and decompression, is considered by Chen to be more important than the SQNR degradation achieved if deconvolution is applied after compression and decompression. In our post-decon compression case, however, predictive deconvolution is followed by different forms of filtering operations, among others multiple attenuation and bandpass filtering. At the same time, these filtering operations are applied *prior to* compression and decompression. Thus, the common offset gathers appear smoother and become much more suited for compression. Therefore, we prefer to employ predictive deconvolution *before* compression and subsequently decompression of the seismic data.

4.11 Conclusions

Lossy data compression makes the storage of seismic data more efficient, and has a potential of reducing the time for network and satellite transmission of seismic data typically from hours to minutes.

The higher the compression ratio is, the more coherent will the compression error appear, consequently implying a stronger influence of compression noise on processing and interpretation. Compression error can be harmful to highly data adaptive processing approaches such as for example predictive deconvolution and common multiple rejection methods (i.e., the parabolic Radon transform). On the other hand, processing steps which reduce ambient noise components by different summation techniques (e.g., migration and stacking) diminish the compression noise.

Important seismic data attributes, including amplitude variation with offset and poststack inversion, for the North Sea data set, are well preserved by the lossy compression method at moderate compression ratios (approximately 7.5:1

for pre-decon compression and 15:1 for post-decon compression).

The seismic industry is careful not to use procedures that introduce noise into the original seismic data set. Nevertheless, speaking about digital seismic data without noise is meaningless due to the sampling of a non-band-limited seismic signal in time and space and subsequently quantization of the amplitudes. Lossy seismic data compression can be seen as just one out of several minor noise sources in the acquisition and processing of seismic data. In best case, seismic data quality can even be increased by correct use of lossy compression since ambient noise components at high frequencies and high wavenumbers are effectively reduced while the dynamic range is well preserved. Given a certain limited resource of storage capacity or transmission bandwidth, lossy seismic data compression is without question a much better alternative than non-sophisticated data reduction methods such as decreasing the temporal range and/or resampling in time, decreasing the spatial range and/or resampling in space, and clipping of the amplitudes.

With an understanding of the associated drawbacks we expect that the need for reduced turnaround times, from seismic data acquisition to processing and interpretation, will push the use of seismic data compression as a routine component of standard survey operations.

4.12 Acknowledgments

We appreciate helpful discussions with our colleagues Andy Morton and Mark Thompson regarding the utilized processing sequence, and we thank Kenneth Duffaut for helping us with the poststack inversion example. Tage Røsten kindly thanks Den norske stats oljeselskap a.s (Statoil) and Petroleum-Geo Services (PGS) for financial support. We also acknowledge Statoil and partners for providing us the North Sea data set and giving us the permission to publish this work.

References

Aase, S. O., 1993, Image subband coding artifacts: Analysis and remedies: Ph.D. thesis, The Norwegian Institute of Technology.

Balasingham, I., 1998, On optimal perfect reconstruction filter banks for image compression: Ph.D. thesis, Norwegian University of Science and Technology.

Barry, K. M., Cavers, D. A., and Kneale, C. W., 1975, Report on recommended standards for digital tape formats: Geophysics, **40**, no. 2, 344–352.

Bosman, C., and Reiter, E. C., 1993, Seismic data compression using wavelet transforms: SEG, 63rd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Washington, D.C., USA, 1261–1264.

Bradley, J., Fei, T., and Hildebrand, S., 1996, Wavelet compression for 3D depth migration: SEG, 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Denver, USA, 1627–1629.

Brislawn, C. M., 1996, Classification of nonexpansive symmetric extension transforms for multirate filter banks: Appl. Comp. Harm. Anal., **3**, no. 4, 337–357.

Buland, A., 1998, Relative amplitude processing – A contractor evaluation: EAGE, 60th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Leipzig, Germany, Session:P004.

Castagna, J. P., and Swan, H. W., 1997, Principles of AVO crossplotting: The Leading Edge, **16**, no. 4, 337–342.

Chen, T., 1995, Seismic data compression: Master's thesis, Center for Wave Phenomena, Colorado School of Mines.

Claypoole, R. L., and Baraniuk, R. G., 1998, Flexible wavelet transforms using lifting: SEG, 68th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, New Orleans, USA, 1979–1982.

Crochiere, R. E., Webber, S. A., and Flanagan, J. L., 1976, Digital coding of speech in sub-bands: Bell Syst. Tech. J., **55**, no. 8, 1069–1085.

Dessing, F. J., and Hoekstra, E. V., 1997, Multiscale tools for seismic data compression:, *in* DELPHI Delft University of Technolog, 8.

Dessing, F. J., and Wapenaar, C. P. A., 1994, Wavefield extrapolation using the wavelet transform: SEG, 64th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Los Angeles, USA, 1355–1358.

Dessing, F. J., and Wapenaar, C. P. A., 1995, Efficient migration with one-way operators in the wavelet transform domain: SEG, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 1240–1243.

Diller, D. E., Hale, D., and Foy, R. D., 1996, Seismic compression exchange standardization via the self-extracting archive model: SEG, 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Denver, USA, 2039–2040.

Donoho, P. L., Ergas, R. A., and Villasenor, J. D., 1995, High-performance seismic trace compression: SEG, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 160–163.

Donoho, P. L., Ergas, R. A., and Polzer, R. S., 1998, Improved data management for interpretation systems using compression: EAGE, 60th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Leipzig, Germany, Session:1–36.

Donoho, P. L., Ergas, R. A., and Polzer, R. S., 1999a, Development of seismic data compression diagnostics: EAGE, 61st Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Helsinki, Finland, Session:P104.

——— 1999b, Development of seismic data compression methods for reliable, low-noise, performance: SEG, 69th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 1903–1906.

Donoho, P. L., Ergas, R. A., and Polzer, R. S., 2000, Diagnostic procedures for safe seismic data compression: EAGE, 62nd Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Glasgow, Scotland, Session:B39.

Duarte, O. d. O., 1992, Predictive deconvolution with additive noise: SEG, 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, New Orleans, USA, 1149–1151.

Duval, L. C., and Røsten, T., 2000, Filter bank decomposition of seismic data with application to compression and denoising: SEG, 70th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Calgary, Canada, 2055–2058.

Duval, L. C., Nguyen, T. Q., and Tran, T. D., 1999a, Seismic data compression and QC using GenLOT: EAGE, 61st Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Helsinki, Finland, Session:P103.

Duval, L. C., Oksman, J., and Nguyen, T. Q., 1999b, A new class of filter banks for seismic data compression: SEG, 69th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 1907–1910.

Ergas, R. A., Polzer, R. S., Donoho, P. L., and Galibert, P.-Y., 1996, Pitfalls in compressing land seismic trace data: EAGE, 58th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Amsterdam, The Netherlands, Session:P156.

Farvardin, N., and Modestino, J. W., 1984, Optimum quantizer performance for a class of non-Gaussian memoryless sources: IEEE Trans. Inform. Theory, **30**, no. 3, 485–497.

Foster, D. J., and Mosher, C. C., 1992, Suppression of multiple reflections using the Radon transform: Geophysics, **57**, no. 3, 386–395.

Foster, D. J., Lane, F. D., Mosher, C. C., and Wu, R.-S., 1997, Wavelet transforms for seismic data processing: SEG, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Dallas, USA, 1318–1321.

Gersho, A., and Gray, R. M., 1991, Vector quantization and signal compression: Kluwer Academic Publishers.

Guo, H., and Burrus, C. S., 1996, Phase-preserving compression of seismic data using the self-adjusting wavelet transform: NASA, Combined Industry, Space and Earth Science Data Compression Workshop, Snowbird, USA, 101–109.

Hall, M., Monk, D., and Reiter, E. C., 1995, An evaluation of seismic data compression on the interpretability of the final product: EAGE, 57th Mtg.

Eur. Assoc. Expl Geophys., Extended Abstracts, Glasgow, Scotland, Session:B035.

Hampson, D., 1986, Inverse velocity stacking for multiple elimination: J. Can. Soc. Expl. Geophys., **22**, no. 1, 44–55.

Hjørungnes, A., and Lervik, J. M., 1997, Jointly optimal classification and uniform threshold quantization in entropy constrained subband image coding: ICASSP, IEEE Proc. Int. Conf. on Acoustics, Speech, and Signal Proc., Munich, USA, 3109–3112.

Hjørungnes, A., Coward, H., and Ramstad, T. A., 1999, Minimum mean square error FIR filter banks with arbitrary filter lengths: ICIP, IEEE Proc. Int. Conf. on Image Proc., Kobe, Japan, 619–623.

Hugonnet, P., and Canadas, G., 1995, Aliasing in the parabolic Radon transform: SEG, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 1366–1369.

Ioup, J. W., and Ioup, G. E., 1998, Noise removal and compression using a wavelet transform: SEG, 68th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, New Orleans, USA, 1076–1079.

Jain, A. K., 1989, Fundamentals of digital image processing: Prentice Hall.

Jayant, N. S., and Noll, P., 1984, Digital coding of waveforms: Prentice Hall.

Junger, A., 1964, Signal-to-noise ratio and record quality: Geophysics, **29**, no. 6, 922–925.

Katto, J., and Yasuda, Y., 1991, Performance evaluation of subband coding and optimization of its filter coefficients: VCIP, Proc. SPIE's Visual Communications and Image Processing, 95–106.

Khène, M. F., and Abdul-Jauwad, S. H., 2000, Efficient seismic compression using the lifting scheme: SEG, 70th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Calgary, Canada, 2052–2054.

Latimer, R. B., Davison, R., and Riel, P. v., 2000, An interpreter's guide to understanding and working with seismic-derived acoustic impedance data: The Leading Edge, **19**, no. 3, 242–256.

Lervik, J. M., and Ramstad, T. A., 1996, Optimality of multiple entropy coder systems for nonstationary sources modelled by a mixture distribution: ICASSP, IEEE Proc. Int. Conf. on Acoustics, Speech, and Signal Proc., Atlanta, USA, 1874–1877.

Lervik, J. M., Røsten, T., and Ramstad, T. A., 1996, Subband seismic data compression: Optimization and evaluation: DSPWS, IEEE Proc. DSP Workshop, Loen, Norway, 65–68.

Lervik, J. M., 1996, Subband image communication over digital transparent and analog waveform channels: Ph.D. thesis, Norwegian University of Science and Technology.

Liner, C. L., 1999, Concepts of normal and dip moveout: Geophysics, **64**, no. 5, 1637–1647.

Luo, Y., and Schuster, G. T., 1992, Wave packet transform and data compression: SEG, 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, New Orleans, USA, 1187–1190.

Malvar, H. S., and Staelin, D. H., 1989, The LOT: Transform coding without blocking effects: IEEE Trans. Acoust., Speech, and Signal Proc., **37**, no. 4, 553–559.

Martucci, S. A., 1991, Signal extension and noncausal filtering for subband coding of images: VCIP, Proc. SPIE's Visual Communications and Image Processing, 137–148.

Meiners, E. P., Lenz, L. L., Dalby, A. E., and Hornsby, J. M., 1972, Recommended standards for digital tape formats: Geophysics, **37**, no. 1, 36–44.

Miao, X., and Cheadle, S., 1998, Noise attenuation with wavelet transforms: SEG, 68th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, New Orleans, USA, 1072–1075.

Mosher, C. C., Keho, T. H., Weglein, A. B., and Foster, D. J., 1996, The impact of migration on AVO: Geophysics, **61**, no. 6, 1603–1615.

Nayebi, K., Barnwell, T. P., and Smith, M. J. T., 1990, The time domain analysis and design of exactly reconstructing FIR analysis/synthesis filter banks: ICASSP, IEEE Proc. Int. Conf. on Acoustics, Speech, and Signal Proc., Albuquerque, USA, 1735–1738.

Nelson, M., and Gailly, J.-L., 1996, The data compression book: M&T Books, second edition.

Northwood, E. J., Weisinger, R. C., and Bradley, J. J., 1967, Recommended standards for digital tape formats: Geophysics, **32**, no. 6, 1073–1084.

Polzer, R. S., Ergas, R. A., Donoho, P. L., and Edmonson, A. L., 1997, Dynamic range characteristics of wavelet data compression - Implications for deconvolution: EAGE, 59th Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Geneva, Switzerland, Session:A039.

Porsani, M. J., and Ursin, B., 1998, Mixed-phase deconvolution: Geophysics, **63**, no. 2, 637–647.

Queiroz, R. L. d., Nguyen, T. Q., and Rao, K. R., 1996, The GenLOT: Generalized linear-phase lapped orthogonal transform: IEEE Trans. Signal Proc., **44**, no. 3, 497–507.

Ramstad, T. A., Aase, S. O., and Husøy, J. H., 1995, Subband compression of images: Principles and examples: Elsevier Science B.V.

Reiter, E. C., and Heller, P. N., 1994, Wavelet transform-based compression of NMO-corrected CDP gathers: SEG, 64th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Los Angeles, USA, 731–734.

Reiter, E. C., 1996, A quantitative comparison of 1, 2 and 3 dimensional wavelet compression methods for seismic data: SEG, 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Denver, USA, 1630–1633.

Rissanen, J., and Langdon, G. G., 1981, Universal modeling and coding: IEEE Trans. Inform. Theory, **27**, no. 1, 12–23.

Ross, C. P., 1993, AVO in the presence of coherent noise: The Leading Edge, **12**, no. 3, 196–201.

Røsten, T., and Ramstad, T. A., 1998, Kirchhoff migration of 2-D post-stack seismic data after subband decomposition: ICIP, IEEE Proc. Int. Conf. on Image Proc., Chicago, USA, 353–357.

Røsten, T., and Ramstad, T. A., 1999, Efficient lossless compression of seismic trace headers using conditional adaptive arithmetic coding: NORSIG, Proc. IEEE 1999 Norwegian Sign. Proc. Symposium, Asker, Norway, 86–90.

Røsten, T., and Waldemar, P., 1998, On the performance of 2-D pre-stack seismic data compression after NMO correction: NORSIG, Proc. 3rd IEEE Nordic Sign. Proc. Symposium, Vigsø, Denmark, 261–264.

Røsten, T., Lervik, J. M., Ramstad, T. A., and Amundsen, L., 1996, Subband compression of seismic stack sections: SEG, 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Denver, USA, 1623–1626.

Røsten, T., Lervik, J. M., Balasingham, I., and Ramstad, T. A., 1997a, On the optimality of filter banks in subband compression of seismic stack sections: SEG, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Dallas, USA, 1338–1341.

Røsten, T., Waldemar, P., Balasingham, I., and Ramstad, T. A., 1997b, A review of subband coding of seismic data: NORSIG, Proc. IEEE 1997 Norwegian Sign. Proc. Symposium, Tromsø, Norway, 69–74.

Røsten, T., Fevang, T., and Ramstad, T. A., 1998a, 2-D post-stack seismic data migration in the subband domain: NORSIG, Proc. 3rd IEEE Nordic Sign. Proc. Symposium, Vigsø, Denmark, 265–268.

Røsten, T., Tegdan, J., and Ramstad, T. A., 1998b, Seismic imaging in the subband domain: ICSPC, IASTED Proc. Int. Conf. on Sign. Proc. and Comm., Las Palmas, Spain, 165–168.

Røsten, T., Marthinussen, V. A., Ramstad, T. A., and Perkis, A., 1999a, Filter bank optimization for high-dimensional compression of pre-stack seismic data: ICASSP, IEEE Proc. Int. Conf. on Acoustics, Speech, and Signal Proc., Phoenix, USA, 3153–3156.

Røsten, T., Ramstad, T. A., and Amundsen, L., 1999b, Seismic data compression, Part I: Subband coding of common offset gathers: Submitted to Geophysics.

——— 1999c, Seismic data compression, Part II: Lossless coding of trace identification headers: Submitted to Geophysics.

Røsten, T., Ramstad, T. A., and Amundsen, L., 2000a, Lossless compression of seismic trace headers: EAGE, 62nd Mtg. Eur. Assoc. Expl Geophys., Extended Abstracts, Glasgow, Scotland, Session:B40.

Røsten, T., Waldemar, P., Buland, A., and Amundsen, L., 2000b, Seismic data compression, Part III: Its influence on processing and interpretation: Submitted to Geophysics.

Russell, B., Hampson, D., and Chun, J., 1990a, Noise elimination and the Radon transform, part 1: The Leading Edge, **9**, no. 10, 18–23.

——— 1990b, Noise elimination and the Radon transform, part 2: The Leading Edge, **9**, no. 11, 31–37.

Russell, B., 1993, Introduction to AVO and this special issue: The Leading Edge, **12**, no. 4, 161.

SEG Tech. Std. Comm., 1997, Digital tape standards – SEG-A, SEG-B, SEG-C, SEG-Y, and SEG-D formats, plus SEG-D revisions 1 and 2: Soc. Expl. Geophys.

Shapiro, J. M., 1993, Embedded image coding using zerotrees of wavelet coefficients: IEEE Trans. Signal Proc., **41**, no. 12, 3445–3462.

Sheriff, R. E., and Geldart, L. P., 1995, Exploration seismology: Cambridge University Press, second edition.

Sheriff, R. E., 1991, Encyclopedic dictionary of exploration geophysics: Soc. Expl. Geophys., third edition.

Shuey, R. T., 1985, A simplification of the Zoeppritz equations: Geophysics, **50**, no. 4, 609–614.

Spanias, A. S., Jonsson, S. B., and Stearns, S. D., 1991, Transform methods for seismic data compression: IEEE Trans. Geosci. Remote Sensing, **29**, no. 3, 407–416.

Stigant, J. P., Ergas, R. A., Donoho, P. L., Minchella, A. S., and Galibert, P.-Y., 1995, Field trial of seismic compression for real-time transmission: SEG, 65th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 960–962.

Stockwell, J. W., 1997, Free software in education: A case study of CWP/SU – Seismic Unix: The Leading Edge, **16**, no. 7, 1045–1049.

Strang, G., and Nguyen, T. Q., 1996, Wavelets and filter banks: Wellesley-Cambridge Press.

Therrien, C. W., 1992, Discrete random signals and statistical signal processing: Prentice Hall.

Ursin, B., and Porsani, M. J., 2000, Estimation of an optimal mixed-phase inverse filter: Geophysical Prospecting, **48**, no. 4, 663–676.

Vassiliou, A., and Wickerhauser, M. V., 1997, Comparison of wavelet image coding schemes for seismic data compression: SEG, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Dallas, USA, 1334–1337.

Vermeer, P., Bragstad, H., and Orr, C., 1996, Aspects of seismic data compression: SEG, 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Denver, USA, 2031–2034.

Vermeer, P., 1999, Compression of field data within system specifications: SEG, 69th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 1911–1913.

Vetterli, M., and Herley, C., 1992, Wavelets and filter banks: Theory and design: IEEE Trans. Signal Proc., **40**, no. 9, 2207–2232.

Villasenor, J. D., Belzer, B., and Liao, J., 1995, Wavelet filter evaluation for image compression: IEEE Trans. Image Proc., **4**, no. 8, 1053–1060.

Villasenor, J. D., Ergas, R. A., and Donoho, P. L., 1996, Seismic data compression using high-dimensional wavelet transforms: DCC, IEEE Data Compression Conference, Snowbird, USA, 396–405.

Waldemar, P., Røsten, T., and Ramstad, T. A., 1997, Compression of seismic stack sections using singular value decomposition: SEG, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Dallas, USA, 1342–1345.

Wang, Y., and Wu, R.-S., 1999, 2-D semi-adapted local cosine/sine transform applied to seismic data compression and its effects on migration: SEG, 69th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Houston, USA, 1918–1921.

Wiggins, J. W., 1988, Attenuation of complex water-bottom multiples by wave-equation-based prediction and subtraction: Geophysics, **53**, no. 12, 1527–1539.

Witten, I. H., Neal, R. M., and Cleary, J. G., 1987, Arithmetic coding for data compression: Comm. of the ACM, **30**, no. 6, 520–540.

Wood, L. C., 1974, Seismic data compression methods: Geophysics, **39**, no. 4, 499–525.

Woods, J. W., and O'Neil, S. D., 1986, Subband coding of images: IEEE Trans. Acoust., Speech, and Signal Proc., **34**, no. 5, 1278–1288.

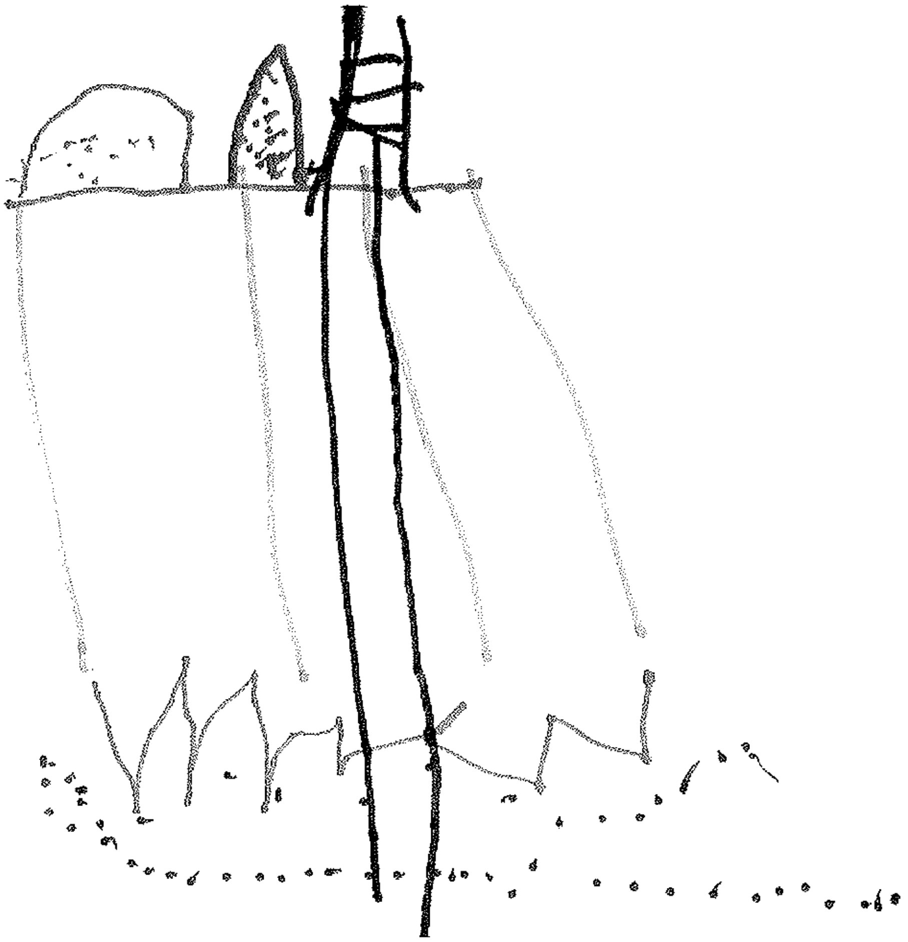
Wu, Y., and McMechan, G. A., 1998, Wave extrapolation in the spatial wavelet domain with application to poststack reverse-time migration: Geophysics, **63**, no. 2, 589–600.

Yilmaz, O., 1987, Seismic data processing: Soc. Expl. Geophys.

Zhou, B., and Greenhalgh, S. A., 1994, Wave-equation extrapolation-based multiple attenuation: 2-D filtering in the f-k domain: Geophysics, **59**, no. 9, 1377–1391.

Index

"Oil platform in the North Sea"



© Courtesy of Stian Røsten