Source–Channel Coding for Wireless Networks

Niklas Wernersson

Communication Theory
School of Electrical Engineering
KTH (Royal Institute of Technology)

Stockholm 2006
Abstract

The aim of source coding is to represent information as accurately as possible using as few bits as possible and in order to do so redundancy from the source needs to be removed. The aim of channel coding is in some sense the contrary, namely to introduce redundancy that can be exploited to protect the information when being transmitted over a nonideal channel. Combining these two techniques leads to the area of joint source–channel coding which in general makes it possible to achieve a better performance when designing a communication system than in the case when source and channel codes are designed separately. In this thesis two particular areas in joint source–channel coding are studied: multiple description coding (MDC) and soft decoding.

Two new MDC schemes are proposed and investigated. The first is based on sorting a frame of samples and transmitting, as side-information/redundancy, an index that describes the resulting permutation. In case that some of the transmitted descriptors are lost during transmission this side information (if received) can be used to estimate the lost descriptors based on the received ones. The second scheme uses permutation codes to produce different descriptions of a block of source data. These descriptions can be used jointly to estimate the original source data. Finally, also the MDC method multiple description coding using pairwise correlating transforms as introduced by Wang et al is studied. A modification of the quantization in this method is proposed which yields a performance gain.

A well known result in joint source–channel coding is that the performance of a communication system can be improved by using soft decoding of the channel output at the cost of a higher decoding complexity. An alternative to this is to quantize the soft information and store the pre-calculated soft decision values in a lookup table. In this thesis we propose new methods for quantizing soft channel information, to be used in conjunction with soft-decision source decoding. The issue on how to best construct finite-bandwidth representations of soft information is also studied.

**Keywords**: source coding, channel coding, joint source–channel coding, multiple description coding, soft decoding, permutation codes.
Acknowledgements

During my work at KTH, which eventually produced this thesis, I have had the pleasure of receiving help from a number of skilful persons. Most valuable, by far, has been all the help from my guiding star Professor Mikael Skoglund. Mikael has been truly generous with his time which is highly appreciated and as a technical adviser I consider him second to none.

I would like to thank the rest of the "plan 4" people including senior researches, computer support, the administrators and finally my Ph.D. student colleges. Also, a special thank to Tomas "II" Andersson who came up with the idea that lead to my first publication. I further wish to thank Dr. Norbert Goertz for taking the time to act as opponent for this thesis.

Finally I would especially like to thank friends and my family (including grandmother Ingeborg and the Brattlôfs; Greta and the Nordlings) for all their support during the years.
# Contents

Abstract i
Acknowledgements iii
Contents v

## I Introduction  1

Introduction 1

1 Source Coding 2
1.1 Lossless Coding 2
1.2 Lossy Coding 3
2 Channel Coding 9
3 The Source–Channel Separation Theorem 10
4 Multiple Description Coding 11
4.1 Theoretical Results 12
4.2 Practical Schemes 13
5 Source Coding for Noisy Channels 19
5.1 Scalar Source Coding for Noisy Channels 20
5.2 Vector Source Coding for Noisy Channels 21
5.3 Index Assignment 23
6 Contributions of the Thesis 23
References 26

## II Included papers 33

A Sorting–Based Multiple Description Quantization A1
1 Introduction A1
2 Sorting-Based MDQ A1
3 Analysis A4
4 Numerical Results A6
5 Conclusions ............................................. A8
Appendix .................................................. A9
References ............................................... A11
Addendum to the Paper ............................... A12

B Multiple Description Coding using Rotated Permutation Codes B1
1 Introduction ......................................... B1
2 Preliminaries ......................................... B2
3 Proposed MDC Scheme ......................... B2
   3.1 Calculating $E[x|i_j]$ .......................... B3
   3.2 Calculating $E[x|i_{1}, \ldots, i_J]$ ............ B4
   3.3 The Effect of the Generating Random Matrices . B6
4 Simulations ......................................... B6
   4.1 Introducing More Design Parameters ....... B7
5 Conclusions ......................................... B8
Appendix A ............................................. B8
Appendix B ............................................. B9
References ............................................... B9

C On Source Decoding Based on Finite-Bandwidth Soft Information C1
1 Introduction ......................................... C1
2 Problem formulation .............................. C2
   2.1 Re-Quantization of a Soft Source Estimate .... C3
   2.2 Vector Quantization of the Soft Channel Values .... C4
   2.3 Scalar Quantization of Soft Channel Values .... C5
3 Implementation of the Different Approaches .... C5
   3.1 Re-Quantization of a Soft Source Estimate .... C6
   3.2 Vector Quantization of the Soft Channel Values .... C6
   3.3 Scalar Quantization of Soft Channel Values .... C7
4 Simulation results .................................. C8
   4.1 SOVQ ............................................. C8
   4.2 COVQ ............................................. C9
5 Conclusions ......................................... C10
References ............................................. C10

D Improved Quantization in Multiple Description Coding by Correlat-
ing Transforms D1
1 Introduction ......................................... D1
2 Preliminaries ......................................... D2
3 Improving the Quantization .................. D4
4 Simulation Results ............................... D6
5 Conclusions ......................................... D7
References ............................................. D8
Part I

Introduction
Introduction

In daily life most people in the western world uses applications resulting from what today is known as the areas of source coding and channel coding. These applications may for instance be compact discs (CDs), mobile phones, MP3 players, digital versatile discs (DVDs), digital television, voice over IP (VoIP), videostreaming etc. This has further lead to a great interest in the area of joint source–channel coding, which in general makes it possible to improve the performance of source and channel coding by designing these basic building blocks jointly instead of treating them as separate units. Source and channel coding is of interest when for instance dealing with transmission of information, i.e. data. A basic block diagram of this is illustrated in Figure 1. Here $X^n = (X_1, X_2, \cdots, X_n)$ is a sequence of source data, originating for instance from sampling a continuous signal, and information about this sequence is to be transmitted over a channel. In order to do so the information in $X^n$ needs to be described such that it can be transmitted over the channel. We also want the receiver to be able to decode the received information and produce the estimate $\hat{X}^n = (\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_n)$ of the original data. The task of the encoder is to produce a representation of $X^n$ and the task of the decoder is to produce the estimate $\hat{X}^n$ based on what was received from the channel. In source–channel coding one is interested in how to design encoders as well as decoders.

![Figure 1: Basic block diagram of data transmission.](image)

This thesis focuses on the area of joint source–channel coding. An introduction to the topic is provided as well as four produced papers (papers A-D). The organization is as follows: Part I contains an introduction where Section 1 explains the
1 Source Coding

When dealing with transmission or storage of information this information generally needs to be represented using a discrete value. Source coding deals with how to represent this information as accurately as possible using as few bits as possible, casually speaking “compression.” The topic can be divided into two cases: \textit{lossless coding}, which requires the source coded version of the source data to be sufficient for reproducing an identical version of the original data. When dealing with \textit{lossy coding} this is no longer required and the aim here is rather to reconstruct an approximated version of the original data which is as good as possible.

How to define “good” is not a trivial question. In source coding this is solved by introducing some structured way of measuring quality. This measure is called \textit{distortion} and can be defined in many ways depending on the context. See for instance [2] for a number of distortion measures applied to gray scale image coding. However, when dealing with the more theoretical aspects of source coding it is well-established practice to use the mean squared error (MSE) as a distortion measure. The dominance of the MSE distortion measure is more likely to arise from the fact that the MSE in many analytical situations can lead to nice and closed form expressions rather than its ability to accurately model the absolute truth about whether an approximation is good or bad. However, in many applications the MSE is a fairly good model for measuring quality and we will in this entire thesis use MSE as a distortion measure. Assuming the vector $X^n$ contains the $n$ source data values $\{X_i\}_{i=1}^n$ and the vector $\hat{X}^n$ contains the $n$ reconstructed values $\{\hat{X}_i\}_{i=1}^n$, the MSE is defined as

$$D_{\text{MSE}} = E \left[ \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{X}_i)^2 \right].$$

(1)

1.1 Lossless Coding

Assume $X^n$, where the $X_i$’s now are discrete values, in Figure 1 describes for example credit card numbers which are to be transmitted to some receiver. In this case it is crucial that the received information is sufficient to extract the exact original data since an approximated value of a credit card number will not be very
useful. This is hence a scenario where it is important that no information is lost when performing source coding which requires lossless coding.

It seems reasonable that there should exist some kind of lower bound on how much the information in \( X^n \) can be compressed in the encoder. This bound does indeed exist and can be found by studying the entropy rate of the process that produces the random vector \( X^n \). Let \( X \) be a discrete random variable with alphabet \( \mathcal{A}_X \) and probability mass function \( p(x) = \Pr(X = x), x \in \mathcal{A}_X \). The entropy \( H(X) \) of \( X \) is defined as

\[
H(X) = - \sum_{x \in \mathcal{A}_X} p(x) \log_2 p(x). \tag{2}
\]

\( H(X) \) is mainly interesting when studying independent identically distributed (i.i.d.) variables. When looking at non–i.i.d. stationary processes the order-\( n \) entropy

\[
H_n(X^n) = - \frac{1}{n} \sum_{x^n \in \mathcal{A}_X^n} p(x^n) \log_2 p(x^n) \tag{3}
\]

and the entropy rate

\[
H_\infty(X) = \lim_{n \to \infty} H_n(X^n) \tag{4}
\]

are of greater interest. Note that all these definitions measures entropy in bits which is not always the case, see e.g. [3]. It turns out that the minimum expected codeword length, \( L_n \), per coded symbol \( X_i \), when coding blocks of length \( n \), satisfies

\[
H_n(X^n) \leq L_n < H_n(X^n) + \frac{1}{n} \tag{5}
\]

meaning that by increasing \( n \), \( L_n \) can get arbitrary close to the entropy rate of a random stationary process. It can be shown that \( H_{n+1}(X^{n+1}) \leq H_n(X^n) \) \( \forall n \) and hence, the entropy rate provides a lower bound on the average length of a uniquely decodable code. For the case of non–stationary processes the reader is referred to [4].

There are a number of coding schemes for performing lossless coding; Huffman coding, Shannon coding, Arithmetic coding and Ziv-Lempel coding are some of the most well known methods [3].

### 1.2 Lossy Coding

As previously stated when dealing with lossy coding we no longer have the requirement of reconstructing an identical copy of the original data \( X^n \). Consider for example the situation when we want to measure the height of a person; the exact length will be a real value meaning that there will be an infinite number of possible outcomes of the measurement. We therefore need to restrict the outcomes somehow, we could for instance assume that the person is taller than 0.5m and no taller than 2.5m. If we also assume that we do not need to measure the length
more accurately than in centimeters the measurement can result in 200 possible outcomes which will approximate the exact length of the person. Approximations like this is done in source coding in order to represent (sampled) continuous signals like sound, video etc. These approximations are referred to as quantization. Furthermore, from (2) it seems intuitive that the smaller the number of possible outcomes, i.e. the course the measurement, the fewer bits are required to represent the measured data. Hence, there exists a fundamental tradeoff between the quality of the data (distortion) and the number of bits required per measurement (rate).

**Scalar/vector Quantization**

Two fundamental tools in lossy coding are scalar and vector quantization. A scalar quantizer is a noninvertible mapping, $Q$, of the real line, $\mathbb{R}$, onto a finite set of points, $\mathcal{C} = \{c_i\}_{i \in \mathcal{I}}$, where $c_i \in \mathbb{R}$ and $\mathcal{I}$ is a finite set of indices,

$$Q : \mathbb{R} \rightarrow \mathcal{C}.$$  \hspace{1cm} (6)

The values in $\mathcal{C}$ constitutes the codebook for $Q$. Assuming $|\mathcal{I}|$ gives the cardinality of $\mathcal{I}$ the quantizer divides the real line into $|\mathcal{I}|$ regions $\mathcal{V}_i$ (some of them may however be empty). These regions are called *quantization cells* and are defined as

$$\mathcal{V}_i = \{x \in \mathbb{R} : Q(x) = c_i\}.$$  \hspace{1cm} (7)

We think of $i$ as the product of the encoder and $c_i$ as the product of the decoder as shown in Figure 2. Vector quantization is a straightforward generalization of scalar quantization to higher dimensions:

$$Q : \mathbb{R}^n \rightarrow \mathcal{C}.$$  \hspace{1cm} (8)

with the modification that $c_i^n \in \mathbb{R}^n$. The quantization cells are defined as

$$\mathcal{V}_i = \{x^n \in \mathbb{R}^n : Q(x^n) = c_i^n\}.$$  \hspace{1cm} (9)

Vector quantization is in some sense the “ultimate” way to quantize a signal vector. No other coding technique exists that can do better than vector quantization for a given number of dimensions and a given rate. Unfortunately the computational complexity of vector quantizers grows exponentially with the dimension making it
infeasible to use unstructured vector quantizers for high dimensions, see e.g. \[5, 6\] for more details on this topic.

Finally we also mention the term Voronoi region: if MSE is used as a distortion measure the scalar/vector quantizer will simply quantize the value \(x^n\) to the closest possible \(c^n_i\). In this case the quantization cells \(V_i\) are called Voronoi regions.

Rate/distortion

As previously stated there seems to be a tradeoff between rate, \(R\), and distortion, \(D\), when performing lossy source coding. To study this we define the encoder as a mapping \(f\) such that

\[
f : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}
\]

and the decoder \(g\)

\[
g : \{1, 2, \ldots, 2^{nR}\} \rightarrow \mathcal{X}^n.
\]

For a pair of \(f\) and \(g\) we get the distortion as

\[
D = E\left[\frac{1}{n}d(X^n, g(f(X^n)))\right]
\]

where \(d(X^n, \hat{X}^n)\) defines the distortion between \(X^n\) and \(\hat{X}^n\) (the special case of MSE was introduced in (1)). A rate distortion pair (\(R, D\)) is achievable if there exist \(f\) and \(g\) such that

\[
\lim_{n \to \infty} E\left[\frac{1}{n}d(X^n, g(f(X^n)))\right] \leq D.
\]

Furthermore, the rate distortion region is defined by the closure of all achievable rate distortion pairs. Also, the rate distortion function \(R(D)\) is given by the infimum of all rates \(R\) that achieves the distortion \(D\). For a stationary and ergodic

![Figure 3: The rate distortion function for zero mean unit variance i.i.d. Gaussian source data.](image-url)
process it can be proved that [3]

\[ R(D) = \lim_{n \to \infty} \frac{1}{n} \inf_{f(\hat{X}^n|X^n) : E(\frac{1}{n}d(X^n, \hat{X}^n)) \leq D} I(X^n; \hat{X}^n). \]  

(14)

This can be seen as a constrained optimization problem: find the \( f(\hat{X}^n|X^n) \) that minimizes mutual information \( I(X^n; \hat{X}^n) \) under the constraint that the distortion is less or equal to \( D \). In Figure 3 the rate distortion function is shown for the well known case of zero mean unit variance i.i.d. Gaussian source data. The fundamental tradeoff between rate and distortion is clearly visible.

**Permutation Coding**

One special kind of lossy coding, which is used in Paper B of this thesis, is permutation coding which will be explained in this section. Permutation coding was introduced by Slepian [7] and Dunn [8] and further developed by Berger [9] which also is a good introduction to the subject. We will here focus on “Variant I” minimum mean-squared error permutation codes. There is also “Variant II” codes but the theory of these is similar to the theory of “Variant I” codes and is therefore not considered.

Permutation coding is an elegant way to perform lossy coding with low complexity. Consider the case when we want to code a sequence of real valued random

![Figure 4](image-url)
variables \( \{X_i\}_{i=1}^{\infty} \). With permutation coding this sequence can be vector quantized in a simple fashion such that the block \( X^n = (X_1, X_2, \ldots, X_n) \) is quantized to an index \( I \in \{1, \ldots, M\} \). To explain the basic idea consider Figure 4 where an experiment where two random vectors \( X^n \) have been generated containing zero mean i.i.d. Gaussian source data. The magnitude of the different samples are plotted on the \( x \)-axis. The first vector has dimension 3 and the second has dimension 101. Furthermore, define \( \xi_j, j = 1, \ldots, n \), to be the \( j \)th largest component of \( X^n \) and then consider the “mid sample,” \( \xi_2 \) in the first plot and \( \xi_{51} \) in the second. If we imagine that we would repeat the experiment by generating new vectors it is clear that \( \xi_{51} \) from this second experiment is likely to be close to \( \xi_{51} \) from the first experiment. This is also true for the the first plot when studying \( \xi_2 \) but we can expect a larger spread for this case. A similar behavior will also be obtained for all the other \( \xi_j \)’s.

Permutation coding uses this fact, namely that knowing the order of the samples in \( X^n \) can be used to estimate the value of each sample. Therefore, the order of the samples is described by the encoder. One of the main advantages of the method is its low complexity, \( O(n \log n) \) from sorting the samples, which makes it possible to perform vector quantization in high dimensions. In [10] it shown that permutation codes are equivalent to entropy coded scalar quantization in the sense that their rate versus distortion relation are identical when \( n \to \infty \). Although this result only holds when \( n \to \infty \) the performance tends to be almost identical as long as intermediate rates are being used for high, but finite, \( n \)’s. For high rates this is no longer true. Typically there exists some level for the rate when increasing the rate no longer improves the performance. This saturation level depends on the
size of \(n\) and increasing \(n\) moves the saturation level to a higher \(R\), see [9]. This is illustrated in Figure 5 where the performances are shown of permutation codes (dashed line), entropy coded quantization (dash–dotted line) as well as the rate distortion function (solid line) for unit variance Gaussian data. For the permutation code \(n = 800\) was used and the saturation effect starts to becomes visible around \(R = 3.5\) bits/sample. In [11] it is shown that, somewhat contrary to intuition, there exist permutation codes with finite \(n\)’s possessing an even better performance than when \(n \to \infty\) and hence also entropy coded scalar quantization. This effect does however depend on the source distribution.

When encoding and decoding in permutation coding there will exist one codeword, for instance corresponding to the first index, of the form

\[
c^n_1 = (\mu_1, \mu_1, \mu_2, \mu_2, \mu_3, \mu_3, \cdots, \mu_K, \mu_K)
\]

(15)

where \(\mu_i\) satisfies \(\mu_1 \leq \mu_2 \leq \cdots \leq \mu_K\) and the \(n_i\)’s are positive integers satisfying \(n_1 + n_2 + \cdots + n_K = n\). All other codewords \(c^n_2, c^n_3, \cdots, c^n_M\) are constructed by creating all possible permutations of \(c^n_1\) meaning that there in total will be

\[
M = \frac{n!}{\prod_{i=1}^{K} n_i!}
\]

(16)

different codewords. If the components of \(X^n\) are i.i.d. all of these permutations are equally likely meaning that the entropy of the permutation index \(I\) will equal \(\log_2 M\). It is a fairly straightforward task to map each of these permutations to a binary number corresponding to a Huffman code. Hence, the rate per coded symbol, \(X_i\), is given from

\[
R \geq \frac{1}{n} \log_2 M
\]

(17)

where “\(\geq\)” means “close to from above”. Also, it turns out that the optimal encoding procedure, for a given set \(\{(n_i, \mu_i)\}_{i=1}^{K}\), is to replace the \(n_1\) smallest components of \(X^n\) by \(\mu_1\), the next \(n_2\) smallest components by \(\mu_2\) and so on. This further means that ordering the components of \(X^n\) also will decide the outcome of the vector quantization. This is an appealing property since sorting can be done with \(O(n \log n)\) complexity which is low enough for implementing permutation vector quantization in very high dimensions.

Now define \(S_i = n_1 + n_2 + \cdots + n_i\) and \(S_0 = 0\) and assume the \(n_i\)’s to be fixed. The optimal choice of \(\mu_i\), for the MSE case, is then given from

\[
\mu_i = \frac{1}{n_i} \sum_{j=S_{i-1}+1}^{S_i} E[\xi_j]
\]

(18)

which can be used to design the different \(\mu_i\)’s. Hence, the expected average of the \(n_1\) smallest \(\xi_j\)’s creates \(\mu_1\) etc. When designing the \(n_i\)’s we instead assume the \(\mu_i\)’s to be fixed. Defining

\[
p_i = \frac{n_i}{n}
\]

(19)
gives an approximation of the rate as

\[ R \approx -\sum_{i=1}^{K} p_i \log_2 p_i. \]  

(20)

Using this as a constraint when minimizing the expected distortion results in a constrained optimization problem giving the optimal choice of \( p_i \) as

\[ p_i = \frac{2^{-\beta \mu_i^2}}{\sum_{j=1}^{K} 2^{-\beta \mu_j^2}} \]  

(21)

where \( \beta \) is chosen such that (20) is valid. However, \( p_i \) will be a real value and \( n_i \) is required to be an integer meaning that we from (21) need to create an approximate optimal value of \( n_i \). With these equations we can optimize (18) and (21) in an iterative fashion eventually converging in some solution for the parameters \( \{\mu_i, n_i\}_{i=1}^{K} \). \( K \) is found by trying out this iteration procedure for different \( K \)'s (many of them can be ruled out) and the best \( K \), i.e. the \( K \) producing the best distortion, is chosen. For a more detailed description of this procedure see [9].

2 Channel Coding

When performing source coding one aims to remove all redundancy in the source data, for channel coding the opposite is done; redundancy is introduced into the data in order to protect the data against channel errors. These channel errors can for instance be packet losses, considered in Papers A, B and D, or bit errors considered in Paper C. In Figure 6 the nonideality of the channel is modelled as discrete memoryless disturbance, \( p(j|i) \), when transmitting the index \( i \) and receiving index \( j \). Note that \( j \) is not necessarily equal to \( i \). Channel coding tries to protect the system against these kinds of imperfections.

![Figure 6: Model of transmission over a channel.](image-url)

There exist theoretical formulas for how much information that, in theory, can be transmitted over a channel with certain statistics. This value is called capacity and tells us the maximum number of bits per channel use that can be transmitted over the channel such that an arbitrary low error probability can be achieved. It should be noted that the capacity is an supremum which it not necessarily achiev-
able itself, this will depend on the channel statistics. For stationary and memory-
less channels the capacity is

\[ C = \max_{p(x)} I(X; Y) \]  

(22)

which is the well known formula for capacity originating from Shannon’s ground-
breaking paper [1]. \( X \) and \( Y \) are not required to be discrete in this formula but
generally when dealing with continuous alphabets a constraint on \( p(x) \) in (22) is
introduced such that the power is restricted, i.e. \( p(x) : E[X^2] \leq P \). Furthermore,
if the channel has memory Dobrushin [12] derived the capacity for “information
stable channels” (see e.g. [13] for explanation) and Verdú and Han [14] showed a
general formula valid for any channel.

3 The Source–Channel Separation Theorem

It is now time to combine the results from the discrete source coding theorem, (5),
and the channel capacity theorem, (22). The discrete source coding theorem states
that the data \( X^n \) can be compressed to use arbitrarily close to \( H_\infty(X) \) bits per
coded source symbol and the channel capacity theorem states that arbitrarily close
to \( C \) bits per channel use can be reliably transmitted over a given channel. Knowing
these separate results the question about how to design the encoder/decoder in a system which needs to do both source and channel coding, as in Figure 6,
arises. Since the discrete source coding theorem only depends on the statistical
properties of the source and the channel coding theorem only depends on the sta-
tistical properties of the channel one might expect that a separate design of source
and channel codes is as good as any other method. It turns out that for stationary
and ergodic sources a source–channel code exist when \( H_\infty(X) < C \) such that
the error probability during transmission can be made arbitrary small. The con-
verse, \( H_\infty(X) > C \), implies that the error probability is bounded away from zero
and it is not possible to achieve arbitrary small error probability. The case when
\( H_\infty(X) = C \) is left unsolved and will depend on the source statistics as well as
the channel properties.

For nonstationary sources the source–channel separation coding theorem takes
an other shape and we need to use concepts like “strictly dominating” and “domi-
nation.” This was introduced and explained in [13, 15].

Based on these theoretical results it may appear as if source and channel codes
could be designed separately. However, this is only true under the assumptions
valid when deriving the results in (5) and (22). One of these assumptions is the
use of infinitely long codes, i.e. \( n \to \infty \). In practice this is not feasible, espe-
cially when dealing with real time applications like video streaming or VoIP. This
motivates the study of joint source–channel coding since for finite \( n \)’s it will be
possible to design better source–channel codes jointly than done separately. This
subject is the main focus in this thesis and in the following sections we will discuss
two particular kinds of joint–source channel coding, namely multiple description coding and source coding for noisy channels.

4 Multiple Description Coding

In multiple description coding (MDC) the total available rate for transmitting source data is split between a number of different channels. Each of these channels may be subject to failure, meaning that some of the transmitted data may be lost. The aim of MDC is then to reconstruct an approximated version of the source data even when only a subset of the used channels are in working state. The problem is illustrated in Figure 7 for two channels. Here $f_1$ and $f_2$ are the encoders used for channels 1 and 2 respectively and defined as

\[ f_k : \mathcal{X}^n \rightarrow \{ 1, 2, \cdots, 2^{nR_k} \} \quad \forall k \in \{ 1, 2 \}. \tag{23} \]

Hence, the encoders will use $R_1$ and $R_2$ of the total available rate $R = R_1 + R_2$. There will exist three decoders: $g_1$ and $g_2$ used when only the information from one channel is received and $g_0$ used when the information from both channels are received, i.e. both channels are in working state. The decoders are defined as

\[ g_k : \{ 1, 2, \cdots, 2^{nR_k} \} \rightarrow \mathcal{X}^n \quad \forall k \in \{ 1, 2 \} \tag{24} \]

\[ g_0 : \{ 1, 2, \cdots, 2^{nR_1} \} \times \{ 1, 2, \cdots, 2^{nR_2} \} \rightarrow \mathcal{X}^n. \tag{25} \]

For the different decoders we define distortions as

\[ D_k = E \left[ \frac{1}{n} d(X^n, \hat{X}_k^n) \right] \quad \forall k \in \{ 0, 1, 2 \}. \tag{26} \]

We call $D_0$ the central distortion and $D_1$ and $D_2$ side distortions.
As an example of MDC, consider the case when $X$ is i.i.d. binary distributed taking values 0 and 1 with probability 1/2. Also assume the Hamming distance is used as a distortion measure, i.e. $d(1, 1) = d(0, 0) = 0$ and $d(0, 1) = d(1, 0) = 1$. Suppose that $D_0 = 0$ is required, $R_1 = R_2 = 0.5$ bits per symbol and the aim is to minimize $D_1 = D_2 = D$. One intuitive approach would be to transmit half of the bits on one channel and the other half on the other channel. This would then give $D = 0.25$ (achieved by simply guessing the value of the lost bits). However, in [16] it is shown that one can do better and it is in fact, somewhat surprisingly, possible to achieve $D = (\sqrt{2} - 1)/2 \approx 0.207$.

The MDC literature is vast, theoretical results as well as practical schemes are presented in the sections below.

4.1 Theoretical Results

One of the first results in MDC was El Gamal and Cover’s region of achievable quintuples $(R_1, R_2, D_0, D_1, D_2)$ [17]. This result states that $(R_1, R_2, D_0, D_1, D_2)$ is achievable if there exist random variables $\hat{X}_0, \hat{X}_1, \hat{X}_2$ jointly distributed with sample $X$ from an i.i.d. source such that

$$R_1 > I(X; \hat{X}_1),$$

$$R_2 > I(X; \hat{X}_2),$$

$$R_1 + R_2 > I(X; \hat{X}_0, \hat{X}_1, \hat{X}_2) + I(\hat{X}_1; \hat{X}_2),$$

$$D_k \leq E[d(X, \hat{X}_k)] \quad \forall k \in \{0, 1, 2\}. $$

Ozarow [18] showed this bound to be tight for the case of Gaussian sources with variance $\sigma_X^2$ (although Ozarow uses $\sigma_X^2 = 1$ in his paper) and also derived closed form expressions for the achievable quintuples which satisfy

$$D_1 \geq \sigma_X^2 e^{-2R_1},$$

$$D_2 \geq \sigma_X^2 e^{-2R_2},$$

$$D_0 \geq \begin{cases} \sigma_X^2 e^{-2(R_1+R_2)} & \text{if } \Pi \geq \Delta \\ \sigma_X^2 e^{-2(R_1+R_2)} & \text{otherwise} \end{cases}$$

where

$$\Pi = (1 - D_1/\sigma_X^2)(1 - D_2/\sigma_X^2)$$

$$\Delta = D_1D_2/\sigma_X^4 - e^{-2(R_1+R_2)}. $$

Studying these equations by setting $R_1 = R_2$ we see that there will be a tradeoff between the performance $D_1, D_2$ versus the performance $D_0$. Decreasing $D_1$ and $D_2$ means that we need to increase $D_0$ and vice versa (can for instance bee seen in Figure 4 of Paper B where $D_1 = D_2$).
Ahlswede [19] showed that the El Gamal–Cover region is tight for the “no excess rate for the joint description” meaning the case when the best possible $D_0$ is achieved according to $R_1 + R_2 = R(D_0)$, where $R(D)$ is the rate distortion formula. In [20] Zhang and Berger constructed a counterexample which shows that the El Gamal–Cover region is not tight in general. The problem of finding a bound that fully describes the achievable multiple description region for two descriptors is still unsolved.

In [21] Zamir shows that

$$D^*(\sigma^2_X, R_1, R_2) \subseteq D_X(R_1, R_2) \subseteq D^*(P_X, R_1, R_2).$$

(36)

Here $\sigma^2_X$ is the variance of the source, $P_X = 2^{2h(X)}/2\pi e$ where $h(X)$ is the differential entropy of $X$. $D^*(\sigma^2, R_1, R_2)$ denotes the set of achievable distortions $(D_0, D_1, D_2)$ when using rates $R_1$ and $R_2$ on a Gaussian source with variance $\sigma^2$. $D_X(R_1, R_2)$ denotes the set of achievable distortions $(D_0, D_1, D_2)$ for the source $X$.

In [22] outer and inner bounds on the achievable quintuples are achieved that relate to the El Gamal–Cover region. The multiple description problem has also been extended the $K$-channel case in [23] as well as in [24, 25] where the area of distributed source coding [26, 27] is used as a tool in MDC. Further results can be found in [28, 29].

### 4.2 Practical Schemes

Also the more practical area of MDC has received considerable attention, see e.g. [30]. Below are a few of the most well known MDC methods explained in brief.

**Multiple Description Scalar Quantizers**

In [31] Vaishampayan makes the first constructive attempt at designing a practical MDC scheme, motivated by the extensive information theory research summarized in the previous section. The paper considers designing scalar quantizers, for memoryless source data, as encoders $(f_1, f_2)$ producing indices $(i, j)$. It is important to note that the quantization intervals of $f_1$ and $f_2$ can be disjoint intervals as shown in Figure 8. This will in turn lead to the existence of a virtual encoder $f_0$ created by the indices from $f_1$ and $f_2$ and an index assignment matrix, see examples in Figure 9. The index generated from $f_1$ is mapped to a row of the index assignment matrix and $f_2$ is mapped to a column. Hence, when both indices are received we know that the original source data must have been in the interval created by the intersection of the two quantization intervals described by $f_1$ and $f_2$, i.e. $x \in \{ x : (f_1(x) = i) \land (f_2(x) = j) \}$. The virtual encoder $f_0$ will therefore give rise to the central distortion $D_0$ which is illustrated in Figure 8 where the left index assignment matrix of Figure 9 is used.
$f_2(x) : 1 \ 1 \ 2 \ 2 \ 3 \ 2 \ x$

$f_1(x) : 1 \ 2 \ 1 \ 2 \ 2 \ 3 \ x$

$f_0(x) : 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ x$

**Figure 8:** Encoders $f_1$ and $f_2$ will together with the (left) index assignment matrix of Figure 9 create a third virtual encoder $f_0$.

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>6</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>21</td>
</tr>
</tbody>
</table>

**Figure 9:** Two examples of index assignment matrices. The left matrix will enable more protection against packet losses and hence a lower performance on $D_0$. The right matrix will on the other hand enable a higher performance on $D_b$.

Based on this idea the MDC system is created by optimizing the lagrangian function

$$L = E[d(X, \hat{X}_0)] + \lambda_1(E[d(X, \hat{X}_1)] - D_1) + \lambda_2(E[d(X, \hat{X}_2)] - D_2).$$

(37)

It is shown that this optimization problem results in a procedure where (i) for a fixed decoder, optimal encoders can be created, and (ii) for a fixed encoder, optimal decoders can be created. Alternating between these two optimization criterions will eventually converge to a solution just as in regular vector quantization training. By choosing low values for $\lambda_1$ and $\lambda_2$ the solution will converge to an MDC scheme with high performance on $D_0$ and low performance on $D_1$ and $D_2$. Choosing high values for the $\lambda_k$’s will on the other hand yield a low performance on $D_0$ and a high performance on $D_1$ and $D_2$. Hence, $\lambda_1$ and $\lambda_2$ can be used to design the system for different levels of error protection.

Furthermore, also the design of the index assignment matrix will impact the tradeoff between $D_0$, $D_1$ and $D_2$. In order to optimize $D_0$ there should be no empty cells in the matrix leading to as many quantization regions for the virtual encoder $f_0$ as possible. On the other hand, if we are interested in only optimizing $D_1$ and $D_2$ there should only be nonempty cells only along the diagonal of the index assignment matrix corresponding to transmitting the same description on
both channels. In Figure 9 two examples of index assignment matrices are shown and since there are more nonempty cells in the right example using this matrix will make it possible to get a better performance on $D_0$ than if the left matrix was used.

The difficult problem on how to actually design the optimal index assignment matrix is not solved in the paper, instead two heuristic methods to design these matrices are presented which are argued to have good performances.

This original idea of Vaishampayan has been investigated and improved in many papers since it was introduced in [31]. In [32] the method is extended to entropy–constrained MDSQ (ECMDSQ). Here, two additional terms are added in the Lagrangian function (37), similar to what is done in entropy–constrained quantization (see e.g. [33]), resulting in

$$L = E[d(X, \hat{X}_0)] + \lambda_1(E[d(X, \hat{X}_1)] - D_1) + \lambda_2(E[d(X, \hat{X}_2)] - D_2)$$

$$+ \lambda_3(H(I) - R_1) + \lambda_4(H(J) - R_2)$$  (38)

where $H(I)$ and $H(J)$ are the entropy of the indices $i$ and $j$ generated by the encoders $f_1$ and $f_2$. It is shown that introducing this modification still leads to an iterative way to optimize the system. Hence, also $\lambda_3$ and $\lambda_4$ can be used to control the convergence of the solution, increasing the values of these will try to force the solution to use a lower rate. A comparison between MDSQ and ECMDSQ can for instance bee seen in Figure 4 of Paper B where $D_1 = D_2$.

In [34] a high–rate analysis of MDSQ and ECMDSQ is presented. Motivated by the fact that comparing different MDC schemes is hard due to the many parameters involved (central/side distortions, rates at the different channels) it is also proposed that the product $D_0D_1$ for the balanced case, i.e. $R_1 = R_2$, is a good figure of merit when measuring performance. As a special case MDSQ/ECMDSQ are analyzed for the Gaussian case and then compared to the Ozarow bound (31-33). This resulted in an an 8.69 dB/3.07 dB gap respectively compared to the theoretical bound. This result was later strengthen in [35].

Some improved results on the index assignment were obtained in [36] and this problem was also studied in [37] where an algorithm is found for designing an index assignment matrix when more than two channels are used.

**Multiple Description Lattice Vector Quantization**

The idea of multiple description lattice vector quantization (MDLVQ) is introduced in [38, 39]. This is in some sense an extension of the idea of MDSQ to the vector quantization case. However, when dealing with unconstrained vector quantization the complexity grows very quickly with the number of dimensions; in order to reduce this complexity the vector quantization can be constrained in some way which generally results in a decreased complexity at the cost of a suboptimal performance. One example of this is lattice vector quantization where all codewords are from a lattice (or possibly a subset). This greatly simplifies the optimal encoding procedure and a lower encoding complexity is achieved. The
high complexity of unconstrained vector quantizers implies that a pure multiple description vector quantization (MDVQ) scheme, suggested in e.g. [40–42], may be impractical for high dimensions and rates which motivates the use of lattice vector quantization in the framework of MDC.

In an \(n\)-dimensional MDL VQ two basic lattices are used: the fine lattice \(\Lambda \subset \mathbb{R}^n\) and the coarser lattice \(\Lambda' \subset \mathbb{R}^n\). \(\Lambda\) will constitute the codewords of the central decoder and \(\Lambda'\) will constitute the codewords of the side decoders. Furthermore, \(\Lambda'\) is chosen such that it is geometrically similar to \(\Lambda\) meaning that \(\Lambda'\) can be created by a rotation and a scaling of \(\Lambda\). In addition, no elements of \(\Lambda\) should lie on the boundaries of the Voronoi regions of \(\Lambda'\). An important parameter is the index

\[
K = \frac{\Lambda}{\Lambda'}
\]

which describes how many lattice points from \(\Lambda\) there exist in the Voronoi regions of \(\Lambda'\) (it is assumed that \(K \geq 1\)). The lower the value of \(K\), the more error protection is put into the system.

Based on these lattices an index assignment mapping function \(\ell\), which is an injection, is created as a one-to-one mapping between a lattice point in \(\Lambda\) and two lattice points in \(\Lambda' \times \Lambda'\). Hence,

\[
\Lambda \xrightarrow{1-1} \ell(\Lambda) \subseteq \Lambda' \times \Lambda'.
\]

The encoder will start quantizing a given vector \(X^n\) to the closest point \(\lambda \in \Lambda\). By deriving \(\ell(\lambda)\) the resulting point is mapped to \((\lambda_1, \lambda_2) \in \Lambda' \times \Lambda'\), i.e. two points in the coarser lattice. It should here be noted that the order of these points are of importance meaning that \(\ell^{-1}(\lambda_1, \lambda_2) \neq \ell^{-1}(\lambda_2, \lambda_1)\). Descriptions of \(\lambda_1\) and \(\lambda_2\) are then transmitted over one channel each and if both descriptions are received the inverse mapping \(\ell^{-1}\) is used to recover \(\lambda\). If only one descriptor is received \(\lambda_1\), or \(\lambda_2\), is used as a reconstruction point. This means that the distance between \(\lambda\) and the \(\lambda_k\)'s will affect the side distortion and [39] considers the design of the index mapping for the symmetric case when producing equal side distortions from equal-rate channels (further investigated in [43] for the asymmetric case). An asymptotic analysis is also provided which reveals that the performance of MDL VQ can get arbitrarily close to the asymptotic multiple description rate distortion bound [44] when the rate and dimension approach infinity. Also [35] provides insight in the asymptotical behavior of MDLQ.

In [45] a simple, yet powerful, modification of the encoder is introduced which makes it possible to not only optimize the encoding after the central distortion which was previously the case. This is done by instead of, as in the original idea, minimizing

\[
\|X^n - \hat{X}_0^n\|^2
\]

minimize

\[
\alpha\|X^n - \hat{X}_0^n\|^2 + \beta(\|X^n - \hat{X}_1^n\|^2 + \|X^n - \hat{X}_2^n\|^2).
\]
Choosing a large value on $\beta$ will decrease the side distortion at the cost of increasing the central distortion and vice versa.

**Multiple Description Using Pairwise Correlating Transforms**

Multiple description coding using pairwise correlating transforms (MDCPC) was introduced in [46–49]. The basic idea here is to create correlation between the transmitted information. This correlation can be exploited in the case of a packet loss on one of the channels since the received packet due to the correlation will contain information also about the lost packet. In order to do this a piecewise correlating transform $T$ is used such that

$$
\begin{bmatrix}
Y_1 \\
Y_2
\end{bmatrix} = T \begin{bmatrix}
X_1 \\
X_2
\end{bmatrix},
$$

(43)

where

$$
T = \begin{bmatrix}
  r_2 \cos \theta_2 & -r_2 \sin \theta_2 \\
- r_1 \cos \theta_1 & r_1 \sin \theta_1
\end{bmatrix}.
$$

(44)

Here $r_1$ and $r_2$ will control the length of the basis vectors and $\theta_1$ and $\theta_2$ will control the direction. The transform is invertible so that

$$
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix} = T^{-1} \begin{bmatrix}
Y_1 \\
Y_2
\end{bmatrix}.
$$

(45)

Based on the choice of $r_1, r_2, \theta_1, \theta_2$ a controlled amount of correlation, i.e. redundancy, will be introduced in $Y_1$ and $Y_2$ which are transmitted over the channels. The more redundancy introduced the lower side distortion will be obtained at the cost of an increased central distortion.

However, in their present form (43)-(45) use continuous values. In order to make the idea implementable quantization needs to be performed at some stage in these equations. This is solved by quantizing $X_1$ and $X_2$ and then finding an approximation of the transform $T$ which ensures that also $Y_1$ and $Y_2$ will be discrete. In Paper D of this thesis we propose to change the order of this such that the transformation is performed first and secondly $Y_1$ and $Y_2$ are quantized. This results in a performance gain.

To get some intuition about the behavior of the original method some of the theoretical results of [48, 50] are reviewed: For the case when $R_1 = R_2 = R$ and using high rate approximations it can be showed that when no redundancy is introduced between the packets, i.e $T$ equals the identity matrix, the performance will behave approximately as

$$
D_0^* = \frac{\pi e}{6} \sigma_1 \sigma_2 2^{-2R}
$$

(46)

$$
D_s^* = \frac{1}{4} \left( \sigma_1^2 + \sigma_2^2 \right) + \frac{\pi e}{12} \sigma_1 \sigma_2 2^{-2R}
$$

(47)
where $D_s$ is the average side distortion between the two channels. Using the transformation matrix

$$T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

(48)

we instead get

$$D_0 = \Gamma D_0^*$$

(49)

$$D_s = \frac{1}{\Gamma^2} \frac{1}{4} (\sigma_1^2 + \sigma_2^2) + \frac{\pi e}{12} \sigma_1 \sigma_2 2^{-2R}$$

(50)

where

$$\Gamma = \frac{(\sigma_1^2 + \sigma_2^2)/2}{\sigma_1 \sigma_2}.$$ 

(51)

Hence, the central distortion is increased (assuming $\sigma_1 > \sigma_2$) and the constant term in the average side distortion is decreased at the same time as the exponential term is increased. Two conclusions can be drawn; firstly MDCPC is not of interest when the rate is very high, since $D_s$ is bounded by a constant term. Secondly, the method also requires unequal variances for the sources $X_1$ and $X_2$ since otherwise $\Gamma = 1$. The method is however efficient in increasing robustness with a small amount of redundancy.

The method has been further developed in [50] where it is extended for using more than two channels. Some analytical results on the case of gaussian source data are presented in [51].

**Multiple Description Coding Using Frames**

The idea of multiple description coding using frames has obvious similarities with ordinary block channel coding, see e.g. [52]. The idea presented in [53–56] is to multiply an $n$-dimensional source vector $X^n$ with a rectangular matrix $F \in \mathbb{R}^{m \times n}$ of rank $n$ and with $m > n$;

$$Y^m = FX^n.$$ 

(52)

$Y^m$ will constitute an overcomplete representation of $X^n$ and $X^n$ is hence described by $m$ descriptors which can be transmitted over one channel each in the ordinary MDC fashion. In the case that at least $n$ descriptors are received $X^n$ can be recovered by creating the (pseudo-)inverse matrix corresponding to the received descriptors in $Y^m$. In the case that $q < n$ descriptors are received the received data will describe and $(n - q)$-dimensional subspace $S_q$ such that $X^n \in S_q$. $X^n$ can then be reconstructed as

$$\hat{x}^n = E[X^n | x^n \in S_q]$$

(53)

if the aim is to minimize the MSE [57].

In the idea above we have not included the fact that quantization will be necessary of the descriptors $Y^m$. The basic idea is however still valid and the impact of the quantization is analyzed in [58].
Other Methods

An other interesting approach was introduced in [59] where entropy–coded dithered lattice quantizers are used. The method is shown to have a good performance but it is not asymptotic optimal. This work is carried on in [60, 61] which results in a scheme that can achieve the whole Ozarow region for the Gaussian case when the dimension becomes large.

In Paper A we study the possibility to use sorting as a tool to produce MDC and somewhat related to this we study the use of permutation codes in an MDC scheme in Paper B.

5 Source Coding for Noisy Channels

In Section 1 we explained the basics of source coding. Here a random variable $X^n$ was encoded into an index $i$ which in turn is decoded to an estimate $\hat{X}_i^n$ of $X^n$. However, when transmitting $i$ over a nonideal channel imperfections in the channel may lead to that we receive a value $j$ which not necessarily equals $i$. This effect needs to be taken into consideration when designing a source code that will be transmitted over a nonideal channel. We introduce the basic concepts by an example; consider a 2–dimensional source distributed as in Figure 10(a) where

![Figure 10](image)

*Figure 10: (a) Illustration if a 2–dimensional distribution and (b) the channel $p(j|i)$."

the source data is uniformly distributed over 3 regions. Further assume that we need to quantize this source to an index $i \in \{1, 2, 3\}$ which will be transmitted over a channel $p(j|i)$ described by Figure 10(b). In traditional source coding we would simply aim for minimizing the distortion without invoking channel knowledge in the design. This would, in the MSE case, produce quantization regions and reconstruction points as illustrated in Figure 11(a). Hence, a large contribution to the distortion occurs when transmitting the index $i = 2$ since the receiver will interpret this as if $i = 1$ was actually transmitted and reconstruct accordingly. However, invoking information about the channel statistics in the design would
make it possible to optimize the system better. This will enable the system to bet-
ter protect itself against channel failures resulting in the design illustrated in Figure
11(b) where a better overall MSE will be obtained than in the previous case. Note
that in the encoding we need to consider both how to design the quantization re-
gions as well as the index assignment, i.e. how to label the quantization regions
with indices (changing the labelling affects the performance). The basic problems
of source coding with noisy channels are clear from this example: quantization,
index assignment and decoding. These problems are discussed in the following
sections.

5.1 Scalar Source Coding for Noisy Channels
The first constructive work on scalar source coding for noisy channels was carried
out by Fine in [62]. Here a communication system with a discrete noisy channel
is considered and rules for creating encoders as well as decoders are presented.
This work continues in [63] where optimal quantizers and decoders are developed
for the case of binary channels, both for the unrestricted scalar quantizer as well
as for uniform quantizers. In [64] Farvardin and Vaishampayan extended this re-
sult to other channels and they also introduced a procedure to improve the index
assignment. These results are central in this topic and are summarized below.

![Diagram of a communication system](image)

**Figure 12:** Model of a communication system.

The considered system is illustrated in Figure 12 where $X$ is assumed to be
an i.i.d. source and $i \in \{1, \cdots, M\}$. It is also assumed that MSE is used as a
distortion measure. For a fixed decoder $g$ the encoder is improved in two steps; first the codewords are relabelled such that

$$E[\hat{X}|I = i_1'] \leq E[\hat{X}|I = i_2'] \leq \cdots \leq E[\hat{X}|I = i_M'].$$  

(54)

This will improve the index assignment although it not necessarily results in the optimal index assignment. Secondly, using this new index assignment the scalar quantizer in $f$ is optimized such that

$$f(x) = \arg \min_i \left( E[(x - \hat{X})^2|I = i] \right).$$  

(55)

This will result in optimal scalar quantizers for a given index assignment and a given decoder $g$ since it will consider the fact that $\hat{X}$ is a random variable. It is further shown that (55) can be developed such that the boarders of the optimal scalar quantizer can be found in an analytical fashion. Altogether, these two steps will improve the performance of the encoder and the algorithm moves on to optimize the decoder under the assumption that the encoder $f$ is fixed. The optimal encoder is given by

$$g(j) = E[X|J = j].$$  

(56)

These equations makes it possible to optimize the encoder and decoder in an iterative fashion and will result in a locally optimal solution for $f$ and $g$, which not necessarily equals the global optimal solution.

### 5.2 Vector Source Coding for Noisy Channels

Early works that can be categorized as vector source coding for noisy channels includes [65, 66] but the first more explicit work can be found in [67]. Here optimality conditions for encoders and decoders are formulated which results in channel optimized vector quantization (COVQ). Also [68, 69] studies this subject where [69] is good introduction to COVQ. The central results of these publications are summarized for the MSE case. The system considered is basically the same as in Figure 12 with $X$ replaced by $X^n$ and $\hat{X}$ replaced by $\hat{X}^n$. For a fixed decoder $g$ the encoder should perform its vector quantization as

$$f(x^n) = \arg \min_i \left( E[(x^n - \hat{X}^n)^2|I = i] \right)$$  

(57)

and for a fixed encoder $f$ the decoder should be designed as

$$g(j) = E[X^n|J = j].$$  

(58)

Also here the design is done by alternating between optimizing the encoder, (57), and the decoder, (58). Note that there is no step for updating the index assignment as in the scalar case but the algorithm usually tends to converge to a good index
assignment. However, the initialization of the index assignment usually affects the performance of the resulting design.

Further work is done in [70] where it is shown that once a COVQ system has been designed the complexity is no greater than ordinary vector quantization. However, just as in regular vector quantization complexity will be an issue when the dimension and/or the rate is high and this is studied in [71] where restricted vector quantizers are designed for noisy channels. These quantizers will in general have a lower performance but also a lower complexity.

**Soft Decoding**

Most of the work done in joint source–channel coding uses a discrete channel model, \( p(j^k|i^k) \), arising from an analog channel in conjunction with a hard decision scheme. This is illustrated in Figure 13 where \( i^k \) is transmitted and distorted by the analog channel such that \( r^k \) is received as

\[
r^k = i^k + w^k
\]

(59)

where \( w^k \) is additive (real–valued) noise. This value is then converted to a discrete value \( j^k \) which results in a discrete channel model. However, if it is assumed that the receiver can access and process the analog (or soft) values \( r^k \) the decoder \( g \) could be based on \( r^k \), instead on \( j^k \). Since \( i^k \to r^k \to j^k \) will constitute a Markov chain we can conclude that \( r^k \) will contain more, or possibly an equal amount of, information about \( i^k \) than \( j^k \) which means that decoding based on \( r^k \) should make it possible to get a better performance than decoding based on \( j^k \).

For instance in the MSE case the optimal decoder is given as

\[
\hat{X}^n = E[X^n|R^k = r^k].
\]

(60)

Decoding based on soft channel values is often referred to as soft decoding and the idea originates from [72]. Further work is the area includes [73] where Hadamard–based optimal and suboptimal soft decoders are developed. In [74] a
general treatment for soft decoding of noisy channels with finite memory is provided. One of the main results is that the complex algorithms resulting from this theory can be written in a recursive manner lowering the complexity. Also in [75] channels with memory is studied.

Using soft decoding results in a high decoding complexity. In [76, 77] this complexity is decreased by quantizing the soft channel values. The quantization produces a vector $j^q_k$ and is hence a compromise between using the complete soft information $r_k$ and the commonly occurring coarse version, $j^k$, of the information. The design criteria used is to design uniform scalar quantizers for the $r_m$’s such that the mutual information between $r_m$ and $j^q_m$ is maximized. We further investigate how to create $j^q_k$ by presenting a different approach in Paper C of this thesis where theory for designing nonuniform quantizers are developed. These quantizers are optimal in the MSE sense.

5.3 Index Assignment

When dealing with noisy channels the index assignment of the different codewords becomes an important issue to consider. Otherwise, codewords that interchange frequently may be far apart in the signal space which may cause a large contribution to the distortion. However, the index assignment problem is NP-hard and to find the optimal solution one would have to run a full search over the $M!$ possible orderings of the $M$ codevectors (although some of these permutations can be ruled out). Conducting a full search is not feasible in most practical situations which have made the research focus on suboptimal algorithms, with limited complexity, to solve the problem.

Early publications about the index assignment problem are [78, 79] where heuristic algorithms are described. In [69, 80] simulated annealing is used to generate good index assignments and in [81] a greedy method called “Pseudo–Gray Coding” is developed which is shown to have a good performance. In [82, 83] the Hadamard transform is used as a tool to create, as well as to indicate the performance of, index assignments for binary symetric channels. The method is demonstrated to have a high performance and a fairly low complexity. This work is extended to cover more general channel models in [84].

6 Contributions of the Thesis

In this thesis we focus on two topics:

- **Multiple description coding.** In papers A and B we introduce new MDC schemes. These schemes are analyzed as well as simulated. In paper D we improve the quantization in multiple description coding using pairwise correlation transforms, briefly explained in Section 4.2.
- *Soft decoding*. The main contribution in paper C is the study of how to quantize soft information for later use in a soft decision scheme.

These papers are summarized in the following sections.

**Paper A: Sorting–based Multiple Description Quantization**

A new method for performing multiple description coding is introduced. The scheme is based on sorting a frame of samples and transmitting, as side-information/redundancy, an index that describes the resulting permutation. In the case that some of the transmitted descriptors are lost this side information (if received) can be used to estimate the lost descriptors based on the received ones. This can be done since the side information describes the order of the descriptors within the frame and hence each received descriptor will narrow down the possible values of the lost ones. The side information is shown to be useful also in the case when no descriptors are lost. For the case of a uniform i.i.d. source a closed form expression for the performance is derived making it possible to analytically optimize the choice of system parameters. Simulations conducted shows the scheme to have a similar performance to multiple description scalar quantization. The main advantage of the suggested method is that it has virtually zero design complexity, making it easy to implement and adapt to varying loss probabilities. It also has the advantage of allowing straightforward implementation of high dimensional MDC.

**Paper B: Multiple Description Coding using Rotated Permutation Codes**

The problem of designing multiple description source codes for $J$ channels is addressed. Our proposed technique consist mainly of two steps; first $J$ copies of the vector $X^n$, containing the source data $\{x_i\}_{i=1}^n$, are rotated by multiplication of $J$ random unitary matrices, i.e. one rotation matrix is generated and used for each channel. Secondly each of the resulting vectors are, independently of the other vectors, vector quantized using permutation source coding. Furthermore, the decoding of the received information when only one channel is received is done by applying the inverse rotation to the result of standard permutation decoding; the optimum strategy for combining the information from multiple channels is a more difficult problem. Instead of implementing optimal combining we propose to simply average the decoded output of the individual channels and then adjust the length of the resulting vector based on a theoretical analysis valid for permutation codes. The choice of using permutation codes comes from the fact that their low complexity makes high dimensional vector quantization possible, i.e. large $n$’s, and our simulations have indicated that the random generation of rotation matrices works well when the dimension is high. For low dimensions different outcomes of the generated rotation matrices seem to yield quite different performance, meaning that the random design may not be as appropriate for this case. Hence, any vector quantization scheme able to perform quantization in high dimensions could potentially replace the permutation coding in the proposed scheme.
Using i.i.d. zero-mean Gaussian data for the $x_i$’s we evaluated the proposed scheme by comparing it to multiple description scalar quantization (MDSQ) as well as entropy-constrained MDSQ (ECMDSQ). It was shown that when using $J = 2$ and $R = 4$ bits/symbol the proposed system outperformed MDSQ. Compared to ECMDSQ a performance gain was achieved when optimizing the systems for receiving, close to, only one descriptor. Otherwise ECMDSQ had the better performance. The main advantages of the proposed method are its relatively low complexity and its ability to easily implement any number of descriptions.

**Paper C: On Source Decoding based on Finite–Bandwidth Soft Information**

Designing a communication system using joint source–channel coding in general makes it possible to achieve a better performance than when the source and channel codes are designed separately, especially under strict delay-constraints. The majority of work done in joint source-channel coding uses a discrete channel model, corresponding to an analog channel in conjunction with a hard decision scheme. The performance of such a system can however be improved by using soft decoding at the cost of a higher decoding complexity. An alternative is to quantize the soft information and store the pre-calculated soft decision values in a lookup table. In this paper we propose new methods for quantizing soft channel information, to be used in conjunction with soft-decision source decoding.

The issue on how to best construct finite-bandwidth representations of soft information is further studied and compared for three main approaches: 1) re-quantization of the soft decoding estimate; 2) vector quantization of the soft channel values, and; 3) scalar quantization of soft channel values. We showed analytically that 1) and 2) are essentially equivalent. Also, since 3) is a special case of 2) it can only yield similar or worse performance. However, we derived expressions that specify the optimal scalar quantizers, and when using designs based on these a performance close to that of approaches 1) and 2) was achieved. The gain of this suboptimality is a substantially lower complexity which makes the method interesting.

**Paper D: Improved Quantization in Multiple Description Coding by Correlating Transforms**

Multiple description using pairwise correlation transforms (MDCPC) is studied. Here a pairwise correlating transforms introduces correlation between different bitstreams. In the case of a lost bitstream, this correlation can be used in order to get an estimate of a lost stream. In this paper we suggest a new approach for performing the quantization in MDCPC. Using the original method the data is quantized and then transformed by a matrix operator in order to increase the redundancy between descriptors. We suggest to reverse the order of these operations: first the data is transformed and then quantized. We show that this leads to a modification of the distortion measure. Using the generalized Lloyd algorithm when designing
the quantization codebook also leads to a new way to update the codevectors. The modification makes it possible to improve the shape of the quantization cells and to tailor these to the employed transform. Our simulations indicates that the modified method performs better than the original one when smaller amounts of redundancy are introduced into the transmitted data. For the simulations conducted the modified method gave 2 dB signal–to–distortion gain compared to the original system when no descriptors were lost. The gain decreased to about 0.5-1 dB when the probability of lost descriptors was increased.

References


