KTH Electrical Engineering

On Error-Robust Source Coding with Image Coding Applications

TOMAS ANDERSSON

Licentiate Thesis in Telecommunications
Stockholm, Sweden, 2006

KTH-Electrical Engineering
TRITA-EE-2006:025 Dept. of Signals, Sensors and Systems
ISSN 1653-5146 SE-100 44 Stockholm, Sweden

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen torsdagen den 15 juni 2006 kl 13 i hörsal Q2, Osquldas väg 10, Stockholm.

Tryck: Universitetsservice US AB

Abstract

This thesis treats the problem of source coding in situations where the encoded data is subject to errors. The typical scenario is a communication system, where source data such as speech or images should be transmitted from one point to another. A problem is that most communication systems introduce some sort of error in the transmission. A wireless communication link is prone to introduce individual bit errors, while in a packet based network, such as the Internet, packet losses are the main source of error.

The traditional approach to this problem is to add error correcting codes on top of the encoded source data, or to employ some scheme for retransmission of lost or corrupted data. The source coding problem is then treated under the assumption that all data that is transmitted from the source encoder reaches the source decoder on the receiving end without any errors. This thesis takes another approach to the problem and treats source and channel coding jointly under the assumption that there is some knowledge about the channel that will be used for transmission. Such joint source–channel coding schemes have potential benefits over the traditional separated approach. More specifically, joint source–channel coding can typically achieve better performance using shorter codes than the separated approach. This is useful in scenarios with constraints on the delay of the system.

Two different flavors of joint source–channel coding are treated in this thesis; multiple description coding and channel optimized vector quantization. Channel optimized vector quantization is a technique to directly incorporate knowledge about the channel into the source coder. This thesis contributes to the field by using channel optimized vector quantization in a couple of new scenarios. Multiple description coding is the concept of encoding a source using several different descriptions in order to provide robustness in systems with losses in the transmission. One contribution of this thesis is an improvement to an existing multiple description coding scheme

and another contribution is to put multiple description coding in the context of channel optimized vector quantization. The thesis also presents a simple image coder which is used to evaluate some of the results on channel optimized vector quantization.

Acknowledgments

The path of writing a thesis is not always a straight road where everything flows without problems. It is full of uphills and downhills and difficult bends every here and there. During times, when it feels like being stuck in an everlasting uphill, it is nice to know that there are people around who will always give a push of encouragement in the right direction. My warmest gratitude goes to all of those who have helped me during the course of writing.

First of all I would like to thank my supervisor Prof. Mikael Skoglund for his professional guidance when the road has been difficult to walk. Without his help, this thesis would never have been written.

Thanks to my fellow coworkers and friends here on the fourth floor. For always providing a good and cheerful atmosphere with a few laughs every now and then. Extra thanks to Niklas W. for joint work in multiple description coding and for proof-reading parts of the thesis. Thanks also to Patrick, Svante and Kalle for additional proof-reading.

Special thanks goes to Dr. Astrid Lundmark for acting as opponent on this thesis.

I would also like to thank my family for always being there for me.

Last but not least I would like to thank Nina for her love, understanding and patience. For the support she gives in more than one way, and for making my life special.

Contents

Chapter 1

Introduction

Today, anywhere we go, any time of day, we are surrounded by electronic devices of various forms and shapes that we use in our daily life. Digital cameras, cellular phones, MP3-players, digital television, IP-phones, videostreaming, etc., are examples of applications that have become more or less commonplace. What these applications all have in common, is that they rely on techniques from the area of *information theory*, an area that was invented by Shannon in 1948 [37]. By tradition, information theory is divided into the areas of source coding and channel coding. However, the trend towards using packet based data networks, such as the Internet, for real-time applications, such as voice over IP, has fueled a great interest in the area of joint source–channel coding.

The topic of this thesis is joint source–channel coding, and as the title implies, the focus is on designing source coders that are robust against transmission errors. Image coding is used as an example application, but the teqniques described are not necessarily limited to image coding.

The first section of this chapter gives an introduction to the basic elements of information theory. The second section provides an outline of the thesis, together with the scientific contributions of this work.

1.1 Source Coding

Source coding comes in two different flavours, lossless and lossy. In both cases the aim is to encode a source into a compact digital representation that can be used for storage or transmission.

Lossless source coding applies to *discrete* sources, where it is important that the encoded source can be decoded completely without errors. The objective is usually to represent, or *compress*, the source, using as few bits as possible while still being uniquely decodable into a perfect replica of the source. Lossless coding is what is used, e.g. in zip-compression of computer files. Lossless coding works by removing statistical dependencies from the source. As a toy example, it is easier to say "ten ones", instead of repeating the word "one" ten times to encode the sequence $\{1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$. The excess information that is removed from the source during the encoding, is termed *redundancy*.

Lossy source coding applies when the need to be able to decode an exact copy can be replaced by a fidelity criterion. Instead of an exact copy of the source, the decoder produces an estimate and the fidelity criterion is a measure of the maximum acceptable deviation between the estimate and the source. The source can either be discrete or continuous-valued. Digital encoding of continuous-valued sources is inherently lossy, since it requires quantization of the source into a discrete representation.

This thesis only treats the case of *lossy* compression. The remainder of this section presents the basic elements of lossy source coding.

1.1.1 Rate–Distortion Theory

In this sub-section we introduce the very basics of a fundamental theory for *source coding subject to a fidelity criterion* [38]. This theory is often called *rate–distortion theory* [5].

Suppose we want to code a sequence $X_1^k = (X_1, \ldots, X_k) \in \mathbb{R}^k$ of samples from a continuous-amplitude stationary and ergodic random process $\{X_n\}$, or a *source*, into a finite-resolution representation

$$\hat{X}_1^k \in \{X_1^k(0), \ldots, X_1^k(M-1)\}.$$

That is, each possible value for the sequence X_1^k is assigned a unique representation $\hat{X}_1^k$ from a set of M possible sequences. Let the *rate* of the representation be

$$R = \frac{\log M}{k} \tag{1.1}$$

(bits per source sample) where 'log' is the binary logarithm. Also, define a (per letter) *distortion* measure $d : \mathbb{R}^2 \to \mathbb{R}_+$, that to each pair X and

$\hat{X}$ assigns a non-negative number $d(X, \hat{X})$, interpreted as the "distance" or "measure of dissimilarity" between X and $\hat{X}$. Furthermore, define the *sequence distortion* d_k as

$$d_k(X_1^k, \hat{X}_1^k) = \frac{1}{k} \sum_{n=1}^{k} d(X_n, \hat{X}_n). \tag{1.2}$$

That is, d_k is the average distance or distortion, per discrete time-instant n, between X_1^k and $\hat{X}_1^k$. Note that for a random source sequence X_1^k, producing a random reproduction sequence $\hat{X}_1^k$, the sequence distortion d_k is a random variable. Therefore we also define the *average (sequence) distortion* between X_1^k and $\hat{X}_1^k$ as

$$\bar{d} = E[d_k(X_1^k, \hat{X}_1^k)]. \tag{1.3}$$

Now, a fundamentally important problem is to study the tradeoff between a low average distortion and a low rate R. This problem is important because in practical applications the process $\{X_n\}$ models the random or unpredictable generation of information from a source, for example, samples from a speech signal, or as studied in this thesis an image. Also, the number R measures the number of bits per source sample that are allocated to code a source sequence into a digital representation, for transmission or storage. Hence, the rate R is tightly related to the bandwidth or storage space that needs to be allocated.

Rate–distortion theory was discovered by Shannon in [37, 38], and this theory characterizes the fundamental tradeoff between rate and distortion. More precisely, for any stationary and ergodic source $\{X_n\}$ there exists a *rate–distortion function* $R(D)$, that measures the minimum possible rate $R = R(D)$ that can support an average distortion D. This result can be formalized as follows. Say that a rate R is *achievable* at distortion D, if it is possible to get $\bar{d} \leq D$ at the rate R. Then, the rate distortion function is defined as

$$R(D) = \inf\{R : R \text{ is achievable at distortion } D\}. \tag{1.4}$$

It is a rather remarkable fact that $R(D)$ can actually be computed, at least in principle, for any stationary and ergodic source model. Specializing, for simplicity, on i.i.d sources, that is, assuming the samples X_ℓ and X_m are

independent for $\ell \neq m$, and equally distributed with probability density function (pdf) f, the rate–distortion function can be computed as

$$R(D) = \min I(X; \hat{X}) \tag{1.5}$$

where the minimum is over all conditional distributions $f(\hat{x}|x)$, subject to

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d(x, \hat{x}) f(\hat{x}|x) f(x) dx d\hat{x} \leq D. \tag{1.6}$$

Also, in (1.5) the entity '$I(X; \hat{X})$' is the *mutual information* between X and $\hat{X}$ assuming the joint distribution $f(x, \hat{x}) = f(\hat{x}|x)f(x)$ for X and $\hat{X}$. That is,

$$I(X; \hat{X}) = \int_{-\infty}^{\infty} f(x) \left\{ \int_{-\infty}^{\infty} f(\hat{x}|x) \log \frac{f(\hat{x}|x)}{f(\hat{x})} d\hat{x} \right\} dx \tag{1.7}$$

where

$$f(\hat{x}) = \int_{-\infty}^{\infty} f(\hat{x}|x) f(x) dx.$$

Through these expressions, we see how $R(D)$ depends on $f(x)$ via the minimization over $f(\hat{x}|x)$ in (1.5).

For a few marginal pdf's $f(x)$ there exist closed form expressions for $R(D)$. For example, for a zero-mean Gaussian $f(x)$ with $\int_{-\infty}^{\infty} x^2 f(x) = \sigma^2$, and using the squared Euclidian distance as the distortion measure, we get

$$R(D) = \frac{1}{2} \log \frac{\sigma^2}{D} \tag{1.8}$$

for all $D \in (0, \sigma^2]$. Note that $R(\sigma^2) = 0$, since the average distortion $\bar{d} = \sigma^2$ can be achieved by always reproducing to $\hat{X}_n = 0$, without transmitting or storing any information about the source sequence. The rate distortion function for the i.i.d Gaussian source is shown in Figure 1.1.

Finally, before closing, we remark that $R(D)$ is always a convex function, and can be inverted to define the *distortion–rate* function $D(R) = R^{-1}(D)$. The function $D(R)$ characterizes the minimum possible average distortion at rate R.

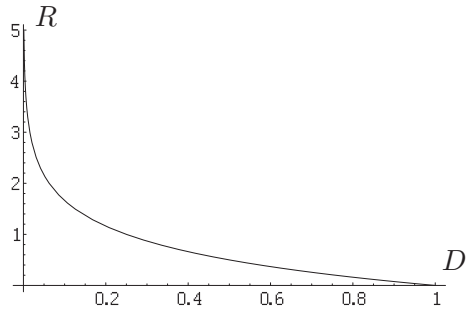


Figure 1.1. Rate–distortion function for an i.i.d. Gaussian source with variance $\sigma^2 = 1$.

1.1.2 Optimal Bit Allocation

Suppose that we have a set of k independent, continuous-valued random variables, $X_1, \ldots, X_k$, that we wish to encode separately, subject to a constraint on the total bit budget. Assume that each X_i is associated with a rate–distortion function $R_i(D_i)$ as discussed in the previous section. Then the problem of optimal bit allocation is that of finding a set of rates $\{R_1, \ldots, R_k\}$ such that $D = \sum D_i$ is minimized, while satisfying the constraint that $\sum R_i \leq R$, where R is the total allowed bit budget.

Using the method of Lagrange multipliers, the optimization problem can be written as

$$\text{minimize } L = \sum D_i + \lambda \sum R_i \tag{1.9}$$

where λ is a positive Lagrange multiplier. Using the distortion–rate function and setting the partial derivatives equal to zero gives

$$\frac{\partial L}{\partial R_i} = \frac{\partial D_i(R_i)}{\partial R_i} + \lambda = 0. \tag{1.10}$$

This means that the optimal solution to the bit allocation problem must satisfy

$$\frac{\partial D_i(R_i)}{\partial R_i} = -\lambda \tag{1.11}$$

for all $i = 0 \ldots k$. Uniqueness follows from the convexity of the rate–distortion curves. When solving (1.11), the value of λ should be selected such that $\sum R_i \leq R$ is satisfied.

The condition (1.11) is called the *equal slope condition* and is a quite intuitive result. Consider the problem of allocating rate for two random variables. Assuming that $R = R_1 + R_2$ is fulfilled, but the slope of $D_1(R_1)$ is steeper than the slope of $D_2(R_2)$. Then adding a small amount of rate to R_1 and removing the same amount of rate from R_2 gives a large decrease of distortion in D_1, but a small increase in D_2. Thus, the overall performance is improved. This can be repeated until the slopes are equal, and it is intuitive that the overall performance can not improve from that point.

1.1.3 Vector Quantization

Here we give a basic introduction to block source coding subject to a distortion criterion or *vector quantization* (VQ). Vector quantization is a general principle for implementing codes that can achieve close to the rate–distortion bounds discussed in Section 1.1.1.

Let $\boldsymbol{X} \in \mathbb{R}^k$ be a k-dimensional random vector,

$$\boldsymbol{X} = \begin{bmatrix} X_1 & X_2 & \cdots & X_k \end{bmatrix}^T$$

drawn according to a pdf $f_{\boldsymbol{X}}(\boldsymbol{x})$. Similarly as in Section 1.1.1, we consider the problem of representing, or quantizing, the possible values for $\boldsymbol{X}$ using a finite set of vectors

$$\mathcal{C} = \{\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{M-1}\}.$$

The set $\mathcal{C}$ is called the *codebook* and its members are called *codevectors* or *codwords*. As illustrated in Figure 1.2, mapping a value $\boldsymbol{x}$ into a codeword $\boldsymbol{c}_i \in \mathcal{C}$ can be described in two steps. Letting

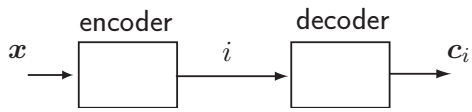


Figure 1.2. Block diagram of vector quantization

$$\mathcal{I}_M = \{0, \ldots, M-1\},$$

the *encoder*, $\varepsilon : \mathbb{R}^k \to \mathcal{I}_M$ takes a realization $\boldsymbol{x}$ for $\boldsymbol{X}$ and maps it into an index $i \in \mathcal{I}_M$. Then the *decoder* $\delta : \mathcal{I}_M \to \mathbb{R}^k$ looks at i and produces the ith codeword $\boldsymbol{c}_i$ in the codebook.

Encoding can be described by the set of *encoder regions*

$$\mathcal{P} = \{\mathcal{S}_0, \ldots, \mathcal{S}_{M-1}\}. \tag{1.12}$$

The encoder regions form a partition of $\mathbb{R}^k$, that is, $\mathbb{R}^k = \bigcup_{i=0}^{M-1} \mathcal{S}_i$ and $\mathcal{S}_i \bigcap \mathcal{S}_j$ is empty for $i \neq j$. Based on the encoder regions, encoding is performed as

$$\boldsymbol{X} \in \mathcal{S}_i \Rightarrow I = i. \tag{1.13}$$

An example of a vector quantizer is illustrated in Figure 1.3. The solid lines represent the encoder partitioning and each cell is assigned to a unique index. The dots correspond to the reconstruction vectors of the decoder codebook.

Designing the encoder, via its associated encoder regions, and the decoder codebook is a special case of the more general design of channel-optimized VQ's discussed in Section 2.2.

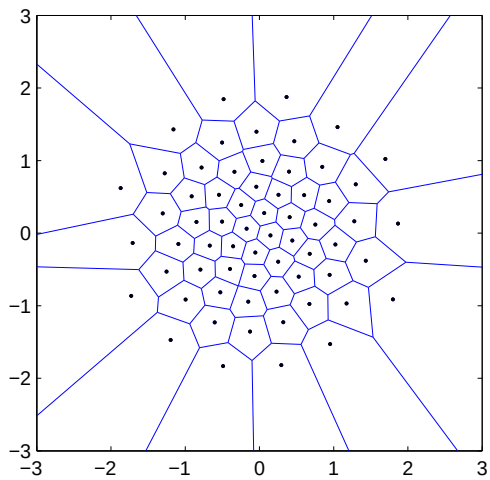


Figure 1.3. A 6 bit 2-dimensional VQ trained for uncorrelated Gaussian data with unit variance. Lines represent decision boundaries and dots represent decoder codewords.

1.1.4 Transform Coding

The optimization procedure in Section 1.1.2 gives the optimal solution only if there is no correlation between the random variables. With correlation present, individual encoding leads to redundancy between the encoded components. On the other hand, vector quantization, as described in the previous section, always distributes the available rate optimally over the k dimensions, regardless of the shape of the joint distribution of the components. However, the complexity of vector quantization grows exponentially with the number of dimensions k, which makes it impractical for sources with many components. This section describes *transform coding*, a useful approach when there is correlation between a large number of random variables that we wish to encode.

Assume as in Section 1.1.3 that we have a vector $\boldsymbol{X} \in \mathbb{R}^k$ consisting of correlated input samples. The idea is then to apply a linear transformation that takes the input vector $\boldsymbol{X}$ and returns a new vector $\boldsymbol{Y}$, also with k components, often referred to as *transform coefficients*. With a suitable choice of the transform, the transform coefficients should be much less correlated than the original input samples. An example is given in Figure 1.4, which illustrates the principle for two dimensional correlated input.

The example shows a large number of realizations of two Gaussian random variables, X_1 and X_2, each with unit variance, $\sigma^2_{X_1} = \sigma^2_{X_2} = 1$, and with covariance $E[X_1 X_2] = 0.9$. After the transformation we get two new Gaussian variables, Y_1 and Y_2, with variances $\sigma^2_{Y_1} = 1.9$ and $\sigma^2_{Y_2} = 0.1$, with zero covariance $E[Y_1 Y_2] = 0$. Using equations (1.8) and (1.11), while keeping a fixed total distortion of $D = 2^{-5}$, it can be shown that the lowest achievable rate when encoding X_1 and X_2 separately is $R = 6$ bits, while the lowest achievable rate when encoding Y_1 and Y_2 separately is $R \approx 4.8$ bits. The improvement corresponds to the amount of statistical redundancy that was removed by the transform.

Note that in the two dimensional example, removing correlation corresponds to a *rotation* of the coordinate axes, which is an operation that can be implemented as a linear transform. In addition, the transform is *orthogonal*, since the new coordinate system has orthogonal coordinates.

In general, for an arbitrary vector dimension k, an *orthogonal transform* can be defined as

$$\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x} \tag{1.14}$$

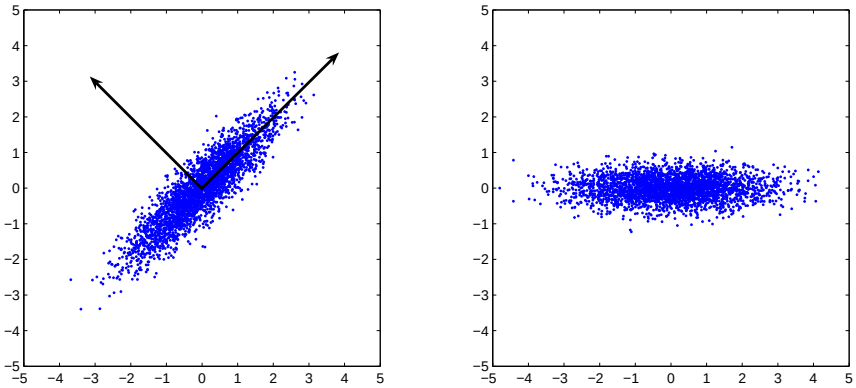


Figure 1.4. Illustrating the principle of transform coding

where $\boldsymbol{T}$ is a real-valued $k \times k$ matrix satisfying the orthogonality constraint

$$\boldsymbol{T}^T = \boldsymbol{T}^{-1} \tag{1.15}$$

or in the complex-valued case

$$\boldsymbol{T}^* = \boldsymbol{T}^{-1} \tag{1.16}$$

where $\boldsymbol{T}^*$ denotes the conjugate transpose of $\boldsymbol{T}$.

Orthogonality of the transform is not an absolute requirement, but has important consequences on the quantization of transform coefficients. The aim of transform coding is to take a vector $\boldsymbol{x}$, transform it into $\boldsymbol{y}$, and then quantize the transform coefficients to obtain $\hat{\boldsymbol{y}}$. Then to reconstruct the source, $\hat{\boldsymbol{x}}$ is formed by taking the inverse transform $\hat{\boldsymbol{x}} = \boldsymbol{T}^{-1}\hat{\boldsymbol{y}}$. This has the side effect that the quantization error $\boldsymbol{y} - \hat{\boldsymbol{y}}$ is multiplied by the inverse transform $\boldsymbol{T}^{-1}$. Thus, the overall distortion is dependent on the transform and obviously not all invertible matrices $\boldsymbol{T}$ are equally suitable for use in transform coding.

Choosing an orthogonal transform, i.e. a transform matrix that satisfies (1.15), has the effect that distances are preserved by the transform. In other words, if $\boldsymbol{y}_1 = \boldsymbol{T}\boldsymbol{x}_1$ and $\boldsymbol{y}_2 = \boldsymbol{T}\boldsymbol{x}_2$, then

$$\|\boldsymbol{x}_2 - \boldsymbol{x}_1\| = \|\boldsymbol{y}_2 - \boldsymbol{y}_1\| . \tag{1.17}$$

To see this, let $\boldsymbol{x} = \boldsymbol{x}_2 - \boldsymbol{x}_1$ and $\boldsymbol{y} = \boldsymbol{y}_2 - \boldsymbol{y}_1$, so that $\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x}$. Then

$$\|\boldsymbol{y}\|^2 = \boldsymbol{y}^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{T}^T\boldsymbol{T}\boldsymbol{x} = \|\boldsymbol{x}\|^2 .$$

This means that any distortion measure that is based on the distance between two points is preserved by the transform. This distance preserving property is sometimes also referred to as the *conservation of energy property*. This is a very useful property in transform coding, since the overall distortion can be described directly from transformed data. The decorrelation property, together with a preserved distortion criterion, makes transform coefficients from an orthogonal transform suitable for separate encoding as described in Section 1.1.2.

1.2 Contributions and Outline

The thesis contributes to the area of robust source coding by two new applications of channel optimized vector quantization (COVQ), an image coder that can benefit from these methods, and by an improvement of an existing multiple description coding scheme. The remainder of this section gives an overview of the outline and points out the contributions of each chapter.

1.2.1 Chapter 2

Chapter 2 contains an overview of important topics in the area of robust source coding. This chapter does not present any new contributions, but is pivotal to the rest of the thesis. First, the concept of joint source–channel coding is explained and motivated. Then, the technique of channel optimized vector quantization is described in detail. Finally, the concepts of index assignment and multiple description coding are described.

1.2.2 Chapter 3

This chapter is related to, but a bit different from the rest of the thesis. It is based on joint work between the author of this thesis and Niklas Wernersson [51][1].

[1]The author of this thesis has changed his last name from Sköllermo to Andersson, so in the reference list, T. Sköllermo is equivalent to T. Andersson

The contribution of this chapter consists of a new approach to perform quantization in a techique called multiple description coding using pairwise correlating transforms, that was originally proposed in [50]. The new technique that is explained in this chapter can be used to reduce the quantization distortion of this multiple description coding scheme.

1.2.3 Chapter 4

The contribution of this chapter is a new image coder [44], which is used to evaluate the robust source coding techniques in this thesis. The image coder is deliberately kept as simple as possible in order to be able to benefit from the robust quantization framework.

First the structure of the image coder is presented. It consists of a subband transform and a vector quantizer for each subband. Section 4.1 describes how the subband transform is constructed from filter banks, and how vectors are selected from each subband.

Next, a model for the probability density functions of the subband vectors is presented. The model relies on assumptions about self similarity of the transform, and is implemented as Gaussian mixture densities.

Finally, the pieces of the image coder are put together and some image examples are given.

1.2.4 Chapter 5

In this chapter we treat situations where both bit errors and erasures are introduced by the channel. Such situations may occur in packet data networks, where part of the transmission is wireless. The contributions consist of constructing a channel model for this type of situation and designing COVQs to operate over these channels [44].

First, the channel model is motivated and presented. Next, it is described how to implement COVQ for this channel. Then the scheme is implemented using the image coder of Chapter 4. Finally, the proposed scheme is compared with using standard VQ, designed without channel knowledge, combined with forward error correction by use of BCH codes.

1.2.5 Chapter 6

Previous chapters of the thesis have discussed channel optimized vector quantization and multiple description coding as two separate approaches to joint

source–channel coding. This chapter contributes by joining the two fields by using the COVQ framework to construct multiple description codes [2].

The chapter starts by defining a channel model to describe the multiple description coding problem. Next, COVQ and index assignment for the multiple description channel model is discussed. Finally, experimental results are presented, which include a comparison between the proposed method and the use of Reed-Solomon codes, and includes some image examples.

1.2.6 Chapter 7

This chapter summarizes the thesis and presents some suggestions of future research.

Chapter 2

Joint Source–Channel Coding

Traditional communication systems separates the two problems of source coding (quantization and/or compression) and channel coding (error protection). The separated approach often simplifies system design and is backed up by Shannon's famous source–channel separation theorem that states that there is no loss in treating the two problems separately. In this chapter we investigate another approach, namely to perform compression and error protection jointly as a single operation. The ideas and methods described in this chapter are by no means novel, and should not be considered as contributions of the thesis. Still, the topic is so central for the thesis that it is worthy a chapter of its own.

First, the source–channel separation theorem is stated more precisely, together with some arguments about its applicability. Then the fundamentals of channel optimized vector quantization are presented in detail. Next, follows a discussion on the problem of index assignment. And finally the idea of multiple description coding is presented.

2.1 Source–Channel Separation Theorem

Here we discuss the fundamental rationale for splitting the problem of digital transmission of analog source data into separate source coding and channel coding, and we discuss under what assumptions such separation can be assumed to be without loss. We focus on discrete-time continuous-amplitude stationary and ergodic sources $\{X_n\}$, like those discussed in Section 1.1.1.

Let the source $\{X_n\}$ have distortion–rate function $D(R)$. Consider encoding k-dimensional sequences from the source at rate R using vector quantization, that is,

$$\boldsymbol{X} = X_{nk+1}^{(n+1)k}, \quad n = 0, 1, \ldots \tag{2.1}$$

is encoded into

$$i = \varepsilon(\boldsymbol{X}) \in \mathcal{I}_M \tag{2.2}$$

by the encoder of a k-dimensional VQ. Assume that $M = 2^{kR}$ is an integer (kR is an integer), then the index i can be described using kR bits. Assume that the kR bits describing i are transmitted over a *noisy binary channel*, by using the channel ρkR times (where $\rho \geq 1$ is chosen such that ρkR is an integer). Since the channel is noisy, received bits need not be equal to transmitted bits. Let $i' \in \mathcal{I}_{M'}$, where $M' = 2^{\rho kR}$, correspond to the bits that are transmitted to represent the information-carrying index i. Since $M' \geq M$, i' is a *redundant* description of i, and the mapping from i to i' is a *channel code*. That is, for each possible index in $\mathcal{I}_M$ there is a corresponding *channel codeword/index* in $\mathcal{I}_{M'}$, and some of the indices in the larger set $\mathcal{I}_{M'}$ are never transmitted. Let $\alpha : \mathcal{I}_M \to \mathcal{I}_{M'}$ describe the channel code, that is, $i' = \alpha(i)$.

For a certain value of i, mapped into i', let $J' \in \mathcal{I}_{M'}$ be the received ρkR-bit (random) index. Since the channel is noisy $\Pr(J' \neq I') > 0$. At the receiver side, the *channel decoder* β maps a realization j' of the received index J' into the most likely information carrying index in $\mathcal{I}_M$. Letting the chosen estimate for the most likely i be denoted j, that is, $j = \beta(j')$, the VQ decoder produces the source vector estimate $\hat{X} = \boldsymbol{c}_j$, the j'th codeword in the VQ codebook.

Let

$$P_e = \Pr(J \neq I) = \sum_{i=0}^{M-1} \Pr(J \neq i | I = i) P(i) \tag{2.3}$$

be the *average error probability* in the channel encoding, transmission and channel decoding. Now, Shannon's *channel coding theorem* [9] states that, as long as $1/\rho < C \leq 1$, where C denotes the *channel capacity* of the binary channel [9], there exists a channel encoder α and a channel decoder β such that P_e is arbitrarily small. More precisely, for a fixed source coding

rate R and ρ, subject to $1/\rho < C$, the error probability P_e can be forced below any $\epsilon > 0$ by choosing a sufficiently large encoding dimension k, and hence also a large resolution M since R is fixed. Furthermore, a (very large) channel code (α, β) that can achieve $P_e < \epsilon$ can be designed without using any knowledge about the source $\{X_n\}$, by assuming that the possible I's are equally likely. The channel capacity C depends only on the random properties of the transmission, and it sets an upper bound on the number of source bits per transmitted channel bit, $1/\rho$, for reliable communication. To set a relative time-reference between the source producing samples and transmitting bits on the channel, assume that the binary channel can be used $\bar{R}$ times per source sample. Since at most a fraction C of the bits transmitted on the channel can be information bits from the VQ encoder, the highest possible source coding rate, in bits per source sample, at which it is still possible to transmit without channel errors, is $R = \bar{R}C$. Consequently the lowest possible distortion is $D(\bar{R}C)$.

It can be proved that the bound $D(\bar{R}C)$ is universal: No matter how the source samples are processed before transmission, it is not possible to achieve a lower distortion, and the bound is determined by $\bar{R}$ and C, which are in turn set by nature. The most important point to make here, is that the optimal distortion $D(\bar{R}C)$ can be achieved by *separate design and implementation* of the source code (mapping $\boldsymbol{X}$ to i) and the channel code (mapping i to i'). However, this separation is in general without loss only in the limit of $k \to \infty$. Under *delay constraints* that prevent the use of a very large dimension k, the separation into source and channel coding can not be assumed to be without loss. In fact, letting the VQ encoder operate on $\boldsymbol{X}$ to produce the higher-resolution description i' directly is in general better than first encoding into i and then using channel encoding to produce i'. This will be discussed further in the following section, and is a central motivation behind the work in this thesis.

A traditional model based on separate source and channel coding is illustrated in Figure 2.1. Here, a vector $\boldsymbol{X}$ is first mapped by a VQ into an index, and this index is then encoded by the encoder, α, of a channel code. In contrast, a system based on joint source–channel coding is illustrated in Figure 2.2. In this system, the vector $\boldsymbol{X}$ is mapped directly into an index for transmission over the channel, by the joint source–channel encoder ε.

Figure 2.1. Traditional model of a communication system.

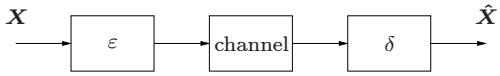


Figure 2.2. Joint source–channel coding model.

2.2 Channel Optimized Vector Quantization

Channel optimized vector quantization (COVQ) is a technique for designing error robust quantizers and originates from work done in the 1980–90's. General results for COVQ are well known [12,27,53], and the basics are repeated in this section.

The principle of channel optimized quantization was first explicitly suggested for scalar quantization in [28]. Another, earlier, reference presenting a strongly related framework is [15]. Farvardin and Vaishampayan extended [28] in several directions, among other things to include the index assignment problem in the design. The first work reported on vector quantizer design for noisy channels is [27]. Another early reference is [53]. The papers that are most often cited for introducing COVQ are however [12,14]. The COVQ concept was generalized in different ways by Farvardin and his students in, for example, [32, 33, 47]. Channel optimized quantization has been applied to image coding, for example in [7,26,40,46]. The papers [7,46] used channel optimized scalar quantization, while [40] used COVQ and [26] trellis coded quantization.

2.2.1 COVQ Basics

Recall from Section 1.1.3 that a vector quantizer is defined by two basic operations, the encoder and decoder. The *encoder*, $\varepsilon(\cdot)$, transforms a source vector, $\boldsymbol{X} \in \mathbb{R}^k$, into a quantization index, $I = \varepsilon(\boldsymbol{X}), \quad I \in \{0, 1, \ldots, M-1\}$. The encoder operation is defined by a partitioning, $\mathcal{P} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{M-1}\}$, of $\mathbb{R}^k$ such that $\varepsilon(\boldsymbol{x}) = i$, iff $\boldsymbol{x} \in \mathcal{S}_i$. The *decoder*, $\delta(\cdot)$, is a mapping from a finite set of integers to an associated set of vectors, $\boldsymbol{Y} = \delta(J), \quad J \in \{0, 1, \ldots, N-1\}$. The set of reconstruction vectors,

$\mathcal{C} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N-1}\}, \quad \boldsymbol{y}_j \in \mathbb{R}^k$, is called the *decoder codebook*. Note that the only difference so far from the ordinary VQ described in Section 1.1.3 is that the size N of the decoder alphabet now is allowed to be different from the size M of the encoder alphabet.

Suppose that the index $I = \varepsilon(\boldsymbol{X})$ is sent over a noisy channel, and that J is observed at the receiver. Assume also that there is a distortion measure $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ associated with mapping an input vector, $\boldsymbol{x}$, into an output vector, $\boldsymbol{y}$. Then the objective is to minimize the expected distortion $D(\mathcal{P}, \mathcal{C}) = E\left[d(\boldsymbol{X}, \boldsymbol{Y})\right]$, where the expectation is to be taken over both the source and the channel distributions. Unfortunately, no closed form solution to this optimization problem exists. Just as in the case of ordinary VQ, we have to treat encoding and decoding separately.

Let $P(j|i) = \Pr(J = j|I = i)$ denote the transition probabilities of the channel. Then the distortion can be written

$$D(\mathcal{P}, \mathcal{C}) = \int_{\boldsymbol{x} \in \mathbb{R}^k} f_{\boldsymbol{X}}(\boldsymbol{x}) \sum_{j=0}^{N-1} P(j|\varepsilon(\boldsymbol{x})) d(\boldsymbol{x}, \boldsymbol{y}_j) d\boldsymbol{x}. \tag{2.4}$$

If the decoder codebook $\{\boldsymbol{y}_j\}_{j=0}^{N-1}$ is fixed, then it is clear from (2.4), that $D(\mathcal{P})$ is minimized if $\sum_{j=0}^{N-1} P(j|\varepsilon(\boldsymbol{x})) d(\boldsymbol{x}, \boldsymbol{y}_j)$ is minimized for each $\boldsymbol{x} \in \mathbb{R}^k$, since $f_{\boldsymbol{X}}(\boldsymbol{x})$ and all terms in the sum are positive. In other words the optimal encoder can be written

$$\varepsilon(\boldsymbol{x}) = \arg\min_i \sum_{j=0}^{N-1} P(j|i) d(\boldsymbol{x}, \boldsymbol{y}_j) \tag{2.5}$$

and the encoder partitioning $\mathcal{P} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{M-1}\}$ is given by

$$\mathcal{S}_i = \left\{ \boldsymbol{x} : \sum_{j=0}^{N-1} P(j|i) d(\boldsymbol{x}, \boldsymbol{y}_j) \leq \sum_{j=0}^{N-1} P(j|i') d(\boldsymbol{x}, \boldsymbol{y}_j), \ \forall i' \neq i \right\}. \tag{2.6}$$

In a similar way, an optimal solution for the decoder can be found. By fixing the encoder the probability $\Pr(J = j)$ of observing a certain channel output is fixed, and the expected distortion with respect to the decoder codebook can be written

$$D(\mathcal{C}) = E\left[d(\boldsymbol{X}, \boldsymbol{Y})\right] = \sum_{i=0}^{N-1} \Pr(J = j) E\left[d(\boldsymbol{X}, \delta(j))|J = j\right]. \tag{2.7}$$

It is clear that $D(\mathcal{P})$ can be minimized by minimizing $E\left[d(\boldsymbol{X}, \delta(j))|J=j\right]$ separately for each j, i.e.

$$\delta(j) = \arg\min_{\boldsymbol{y}_j} E\left[d(\boldsymbol{X}, y_j)|J=j\right]. \tag{2.8}$$

In the special case that the distortion measure is the squared Euclidean distance, $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|^2$, the solution to (2.8) follows from elementary estimation theory and is given by

$$\boldsymbol{y}_j = E\left[\boldsymbol{X}|J=j\right]. \tag{2.9}$$

The expressions for the encoder and decoder given in (2.5) and (2.8) are *necessary* but not sufficient for an optimal encoder–decoder pair. This means that the optimal solution must satisfy (2.5) and (2.8), but fulfilling them does not guarantee the globally optimal solution.

2.2.2 Implementing the Encoder

It might seem that the complexity of channel optimized vector quantization is much higher than for ordinary vector quantization. This is true when speaking of the initial design and training of COVQ, but design and training is usually performed off line. As we shall see, using predesigned COVQ's in a real system requires no more complexity than a normal VQ, assuming that the distortion measure is the squared norm of the error.

Consider the case of the encoder. The expression in (2.5) can be written as follows:

$$\arg\min_i \sum_{j=0}^{N-1} P(j|i)\left\|\boldsymbol{x}-\boldsymbol{y}_j\right\|^2 = \arg\min_i E\left[\left\|\boldsymbol{x}-\boldsymbol{y}_j\right\|^2 |I=i\right]. \tag{2.10}$$

Expanding this expression gives

$$\begin{aligned} E\left[\left\|\boldsymbol{x}-\boldsymbol{y}_j\right\|^2 |I=i\right] &= E\left[\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{y}_j + \boldsymbol{y}_j^T\boldsymbol{y}_j|I=i\right] \\ &= \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T E\left[\boldsymbol{y}_j|I=i\right] + E\left[\boldsymbol{y}_j^T\boldsymbol{y}_j|I=i\right]. \end{aligned}$$

Now introduce $\boldsymbol{v}_i = E\left[\boldsymbol{y}_j | I = i\right]$ and $s_i = E\left[\boldsymbol{y}_j^T \boldsymbol{y}_j | I = i\right]$. These values can be calculated off line and stored in tables at the encoder side. Together, they play the role of an "encoder codebook" and (2.5) simplifies to

$$\varepsilon(\boldsymbol{x}) = \arg\min_i \left(s_i - 2\boldsymbol{x}^T \boldsymbol{v}_i\right). \tag{2.11}$$

The computational complexity of (2.11) is equal to the computational complexity of a normal VQ.

An interesting observation can be made if the input vector $\boldsymbol{x}$ is augmented with a zero and $\boldsymbol{v}_i$ is augmented with $\tilde{s}_i = \sqrt{s_i - \boldsymbol{v}_i^T \boldsymbol{v}_i}$, i.e.

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ 0 \end{bmatrix}, \quad \tilde{\boldsymbol{v}}_{\boldsymbol{i}} = \begin{bmatrix} \boldsymbol{v}_i \\ \tilde{s}_i \end{bmatrix}.$$

This means that (2.5) can be written

$$\varepsilon(\boldsymbol{x}) = \arg\min_i \ \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{v}}_{\boldsymbol{i}}\|^2 \tag{2.12}$$

and that the sets of the encoder partitioning are on the form

$$\mathcal{S}_i = \left\{ \boldsymbol{x} : \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{v}}_{\boldsymbol{i}}\|^2 \leq \|\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{v}}_{\boldsymbol{i}}\|^2, \ \forall i' \neq i \right\}. \tag{2.13}$$

The consequence of (2.13) is that the quantization regions $\mathcal{S}_i$ have the shape of Voronoi regions in a space with dimension $k+1$, where the input space is constrained to a hyper-plane in k dimensions. This means that all fast search methods designed for standard VQ that are based on this structure can also be used for COVQ. (2.13) also allows some insight into the way that redundancy is added in a COVQ system. The term $\tilde{s}_i$ is a measure of the expected distortion associated with coding an input vector into the index i. If the value of $\tilde{s}_i$ is large, then the center of $\mathcal{S}_i$ will be pushed away from the input space, making the intersection between $\mathcal{S}_i$ and the input space smaller. The result is that the probability of an input vector being encoded as i becomes smaller. That this effect introduces statistical redundancy to the system is most obvious in the case that $\tilde{s}_i$ is large enough to push $\mathcal{S}_i$ completely away from the input space, meaning that no input vectors will be encoded as i.

2.2.3 Implementing the Decoder

In a real system, the decoder is simply implemented as a look up table and all that needs to be done is to store the values $\{\boldsymbol{y}_j\}_{j=0}^{N-1}$. During the training procedure, the values of $\boldsymbol{y}_j$ have to be calculated using (2.9). With a slight abuse of notation, let $P(i) = \Pr(I = i)$, $P(i|j) = \Pr(I = i|J = j)$, etc. Then,

$$\begin{aligned} E\left[\boldsymbol{X}|J=j\right] &= \int_{\boldsymbol{x}\in\mathbb{R}^k} \boldsymbol{x} f_{\boldsymbol{X}|J}(\boldsymbol{x}|j) d\boldsymbol{x} \\ &= \sum_{i=0}^{M-1} \int_{\boldsymbol{x}\in\mathcal{S}_i} \boldsymbol{x} f_{\boldsymbol{X}|J}(\boldsymbol{x}|j) d\boldsymbol{x} \end{aligned}$$

using Bayes' rule to replace $f_{\boldsymbol{X}|J}(\boldsymbol{x}|j) = \frac{f_{\boldsymbol{X}}(\boldsymbol{x})P(j|\boldsymbol{x})}{P(j)}$ gives

$$E\left[\boldsymbol{X}|J=j\right] = \sum_{i=0}^{M-1} \int_{\boldsymbol{x}\in\mathcal{S}_i} \boldsymbol{x} \frac{f_{\boldsymbol{X}}(\boldsymbol{x})P(j|\boldsymbol{x})}{P(j)} d\boldsymbol{x}$$

but $P(j|\boldsymbol{x}) = P(j|i)$ for all $\boldsymbol{x} \in \mathcal{S}_i$

$$E\left[\boldsymbol{X}|J=j\right] = \sum_{i=0}^{M-1} \int_{\boldsymbol{x}\in\mathcal{S}_i} \boldsymbol{x} \frac{f_{\boldsymbol{X}}(\boldsymbol{x})P(j|i)}{P(j)} d\boldsymbol{x}$$

and $f_{\boldsymbol{X}}(\boldsymbol{x}) = P(i) f_{\boldsymbol{X}|I}(\boldsymbol{x}|i)$

$$E\left[\boldsymbol{X}|J=j\right] = \sum_{i=0}^{M-1} \frac{P(i)P(j|i)}{P(j)} \int_{\boldsymbol{x}\in\mathcal{S}_i} \boldsymbol{x} f_{\boldsymbol{X}|I}(\boldsymbol{x}|i) d\boldsymbol{x}.$$

Finally we can write the expression of the decoder

$$\boldsymbol{y}_j = E\left[\boldsymbol{X}|J=j\right] = \frac{\sum_{i=0}^{M-1} P(i)P(j|i)\boldsymbol{c}_i}{\sum_{i=0}^{M-1} P(i)P(j|i)}, \tag{2.14}$$

where $\boldsymbol{c}_i = \int_{\boldsymbol{x}\in\mathcal{S}_i} \boldsymbol{x} f_{\boldsymbol{X}|I}(\boldsymbol{x}|i) d\boldsymbol{x} = E\left[\boldsymbol{X}|I=i\right]$, defines the *encoder centroids*.

2.2.4 Training

Generally [12, 27, 53], COVQ design is based on iterating between (2.10) and (2.14) until convergence to a (local) optimum in terms of a stationary point of $D(\mathcal{P}, \mathcal{C})$.

The main problem in the training procedure is to calculate the values of $\{P(i)\}_{i=0}^{M-1}$ and $\{\boldsymbol{c}_i\}_{i=0}^{M-1}$. The exact distribution of $\boldsymbol{X} \in \mathbb{R}^k$ might not be known, and even if it were, the integration would become very tedious when the number of VQ dimensions k is larger than one. The solution normally taken, is to perform stochastic integration based on a training set, $\{\boldsymbol{x}_l\}_{l=0}^{L-1}$, consisting of a large number of samples of $\boldsymbol{X}$. By applying (2.10) to all samples in the training set, $P(i)$ can be estimated from the number of samples that are encoded as i and $\boldsymbol{c}_i$ is taken to be the sample mean of those samples.

The training starts by selecting an initial codebook. This can be for instance the decoder codebook of a VQ trained for the same source and rate. Then all training vectors are quantized using (2.10). This gives the estimates for $P(i)$ and $\boldsymbol{c}_i$, which can then be used to update the decoder codebook $\{\boldsymbol{y}_j\}_{j=0}^{N-1}$ by using (2.14). This procedure is iterated until a certain stopping criterion is met, e.g. the improvement in distortion between two iterations is below some given threshold. The procedure is summarized in Table 2.1.

Table 2.1. Design steps in COVQ generation

1. Select training set and initial codebook
2. Quantize training set using (2.10)
3. Estimate $P(i)$ and $\boldsymbol{c}_i$ from result of quantization
4. Update the decoder codebook $\boldsymbol{y}_j$ using (2.14)
5. Has the training converged? If not goto step 2

2.3 Index Assignment

Here we describe the *index assignment* (IA) problem in connection to VQ over noisy channels. The IA problem is sometimes considered as an integral part of quantizer design for noisy channels, like in [13], however most often it is studied as a separate problem. One of the first studies of the IA problem for (scalar) quantization over a discrete noisy channel was presented in [45]. Other important contributions are included in [12, 24, 54].

To describe the IA problem, consider a VQ (designed assuming noiseless transmission) described by the encoder regions $\{\mathcal{S}_i\}_{i=0}^{M-1}$ and codewords $\{\boldsymbol{c}_i\}_{i=0}^{M-1}$. Assume that after encoding a random vector $\boldsymbol{X}$ to an index i,

$$\boldsymbol{X} \in \mathcal{S}_i \implies I = i, \tag{2.15}$$

the integer i is transmitted in binary format over a binary channel that introduces bit-errors. At the receiver side, the received bits are mapped into an index j, and the decoder outputs $\boldsymbol{c}_j$ as an estimate for $\boldsymbol{X}$. Since there may be bit-errors in the transmission, the event $j \neq i$ has a non-zero probability. Assuming, for simplicity, that the binary channel is memoryless, the event that there is one bit-error in j is more likely than the event that there are more than one error. Hence, if there is a transmission error, received indices, j, that differ in only one bit are the most likely to be received.

Figure 2.3 illustrates a $k = 2$ dimensional size $M = 2^3 = 8$ VQ. The black dots are the codewords, and the boundaries of the encoder regions are marked by solid lines. As can clearly be seen in the figure, one bit-error can lead to quite different quantization-and-channel-noise distortion. More precisely, assume integers are mapped to binary words using the natural binary code ($0 \to 000$, $1 \to 001$, etc.), and assume the correct index is $i = 0$. If there is a transmission error, $j = 1$ is one of the three most likely received indices. As illustrated, the error $i = 0 \to j = 1$ gives a "small" distortion, in this example. However, assuming instead the correct index is $i = 7$, then $j = 6$ is one of the most likely received indices, if there is an error. As can be seen, the error $i = 7 \to j = 6$ gives a larger distortion than the error $i = 0 \to j = 1$!

In general, the problem of mapping codewords in a VQ to indices in order to minimize the average distortion with respect to quantization noise and transmission errors is NP complete [24]. The fundamental problem in IA design is that assigning an index to a codeword constrains the assignment of indices to all the other codewords, since the same index cannot be used

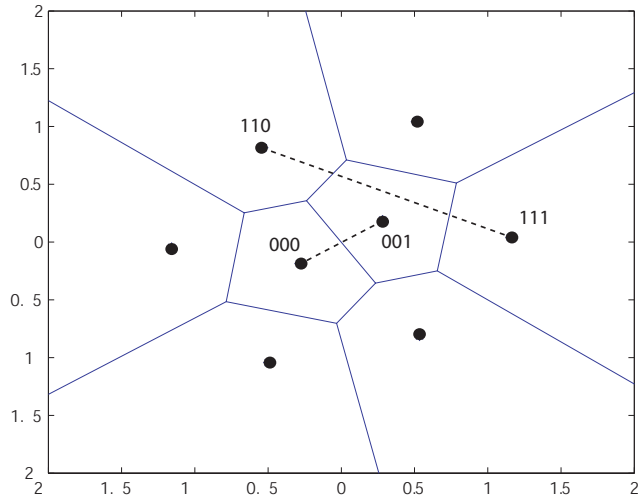


Figure 2.3. Illustrating the IA problem.

again. As indices are assigned, the constraint hardens, and it is therefore very hard to come up with an assignment that is "uniformly good" for all codevectors.

Since the IA problem is NP complete in general, there have been many suggestions for sub-optimal but useful algorithms in the literature. For vector quantization, one of the first studies appears in [54], where a simple algorithm based on flipping bits was presented. Another early, and often cited, study is the one in [12]. The IA algorithm in [12] was based on simulated annealing. Another interesting approach was suggested in [24], utilizing the Hadamard transform as a tool to analyze the impact of bit-errors in the transmission. This method was generalized in [41].

As mentioned, the IA problem is often treated separately from the COVQ design problem. In principle, however, COVQ design includes the IA problem, since the necessary conditions presented in Section 2.2.1 depend on the assignment of indices to encoder regions and codevectors. Therefore, an optimal COVQ design also gives an optimal IA, for the assumed channel model. This fact has been utilized by some authors to implement a good IA, see for example [16], by training a COVQ assuming a high bit-error probability,

enforcing a good IA, and then relaxing the assumed error probability to produce a COVQ with better source coding performance and an inherent good IA.

2.4 Multiple Description Coding

The basic principle of multiple description coding (MDC) is to encode the source into several different *descriptions*. The different descriptions are then transmitted over different channels. The idea is that the decoder should be able to form an estimate of the source even if only a subset of the descriptions is received. A typical feature of multiple description coding is that each description by itself should present the decoder with enough information to decode an estimate of the source. In addition, descriptions should add constructively, in the sense that receiving more descriptions should increase the performance of the estimate.

The most studied multiple description coding scenario is the two channel case depicted in Figure 2.4. We will use this figure to illustrate the basic principle of MDC. Later, in Chapters 3 and 6, we will return to this problem and discuss it in more detail.

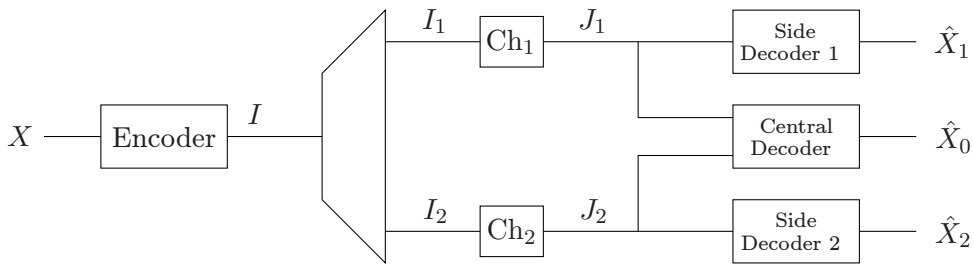


Figure 2.4. Two channel multiple description coding scheme.

Figure 2.4 illustrates the MDC problem for scalar quantization and two descriptions. A source sample X is encoded and transmitted via two different channels, to produce the three different estimates $\hat{X}_i$, $i = 0, 1, 2$, at the receiver. In the classical MDC problem [17, 48], the channels either work perfectly or any of them, or both, are completely defect. Whether a channel works or is defect is known at the receiver side. The principle of MDC can be said to be the production of *diversity* against the event that one or

several channels break down. In modern applications of MDC, a "channel" is often associated with a "packet" in packet-based transmission, and the event "defective channel" is then the same as "packet loss."

Consider Figure 2.4, and let $M = M_1 M_2$. The encoder of the MDC system maps X into an index $I \in \mathcal{I}_M$. This index is then split into *two different descriptions*, $I_1 \in \mathcal{I}_{M_1}$ and $I_2 \in \mathcal{I}_{M_2}$, for example (but not necessarily) via the relation

$$I = I_1 + I_2 M_1. \tag{2.16}$$

The index I_1 is transmitted over channel 1, and I_2 is transmitted over channel 2. Channel 1 either works perfectly, $J_1 = I_1$, or does not work (no J_1 received). The same holds for Channel 2. Hence the possible received information is 'nothing,' (I_1,'nothing'), ('nothing', I_2) or (I_1, I_2). As illustrated, these four possibilities are mapped to $E[X]$, $\hat{X}_1$, $\hat{X}_2$, and $\hat{X}_0$, respectively. Loosely stated, a good MDC should work such that $\hat{X}_i$, $i = 0, 1, 2$, are all useful. This is in contrast to, for example, a multi-resolution code, where one of the descriptions adds constructively to the other but is not useful on its own.

A MDC can be designed in different ways. In this thesis, we will investigate two fundamentally different approaches to the design problem. In Chapter 3, we use linear correlating transforms and in Chapter 6 we extend the COVQ framework to hold for the case of multiple descriptions. A generic MDC design problem can be stated as follows (see, e.g., [48]): Given M_1 and M_2 (the rates that can be used on the two channels), minimize

$$E[d_0(X, \hat{X}_0)] \tag{2.17}$$

subject to

$$E[d_1(X, \hat{X}_1)] \leq D_1, \quad E[d_2(X, \hat{X}_2)] \leq D_2. \tag{2.18}$$

Here, d_i, $i = 0, 1, 2$, are distortion measures. That is, the problem is to minimize the average *central distortion* $E[d_0(X, \hat{X}_0)]$ subject to constraints on the average *side distortions*. This constraint is needed, since simultaneous minimization of the central distortion and side distortions are obviously conflicting goals.

Chapter 3

Improved Quantization in Multiple Description Coding by Correlating Transforms

3.1 Introduction

Packet networks have gained in importance in recent years, for instance by the wide-spread use of the Internet. By using these networks large amounts of data can be transmitted. When transmitting for instance an image a current network system typically uses the TCP protocol to control the transmission as well as the retransmission of lost packages. Unfortunately, packet losses can in general not be neglected and this problem therefore has to be considered when constructing a communication system. The compression algorithms in conventional systems quite often put quite a lot of faith into the delivery system which gives rise to some unwanted effects.

Suppose that N packets are used to transmit, for example, a compressed image and the receiver reconstructs the image as the packets arrive. A problem would arise if the receiver is dependent on receiving all the previous packets in order to reconstruct the data. For instance if packets $\{1, 3, 4...N\}$ are received it would be an undesirable property if only the information in packet 1 could be used until packet 2 eventually arrives. This would produce delays in the system and great dependency on the retransmission process. In the case of a real time system the use of the received packets may have been in vain because of a lost packet. As described in Chapter 2, Section 2.4, one way

to deal with this is to use *multiple description coding*, where each received packet will increase the quality of the image no matter which other packets that have been received. We discussed the basics of MDC in Chapter 2, and some relevant references are [10, 19, 20, 25, 34, 48–50].

In this chapter a new approach to MDC using *pairwise correlating transforms* is presented. In previous work, e.g. [50], the data is first quantized and then transformed. We suggest to reverse the order of these operations, leading to performance gains. The optimal cell shape of the transformed data relates to the optimal cell shape of the original data through some basic equations which makes it possible to perform quantization and designing the codewords after the data is transformed. Only the case with two descriptors will be considered but the theory can easily be extended to handle more descriptors. It is assumed that only one descriptor can be lost at a time (not both) and that the receiver knows when a descriptor is lost. The two channels are also assumed to have equal failure probability, p_{error}, and MSE is used as a distortion measure. The source signal is modeled as uncorrelated Gaussian distributed.

This chapter is organized as follows. In Section 3.2 some preliminary theory of MDC using pairwise correlating transforms is discussed. In Section 3.3 the new approach for MDC using pairwise correlating transforms is presented. In Sections 3.4 and 3.5 some results and conclusions will be presented.

3.2 Preliminaries

Generally the objective with transform coding is to remove redundancy in the data in order to decrease the entropy. The goal of MDC is the opposite, namely to introduce redundancy in the data but in a controlled fashion. A quite natural approach for this is to first remove possible redundancy in the data by for instance using the Karhunen-Loeve transform. After this MDC is used in order to introduce redundancy again, but this time in selected amounts. In this chapter it is assumed that the original data is uncorrelated Gaussian distributed so the problem of removing initial redundancy will not be considered.

In Figure 3.1 the basic structure of the MDC described in [50] is shown. The data variables A and B are to be transmitted and are quantized into $\overline{A}$ and $\overline{B}$. These values are then transformed using the transform

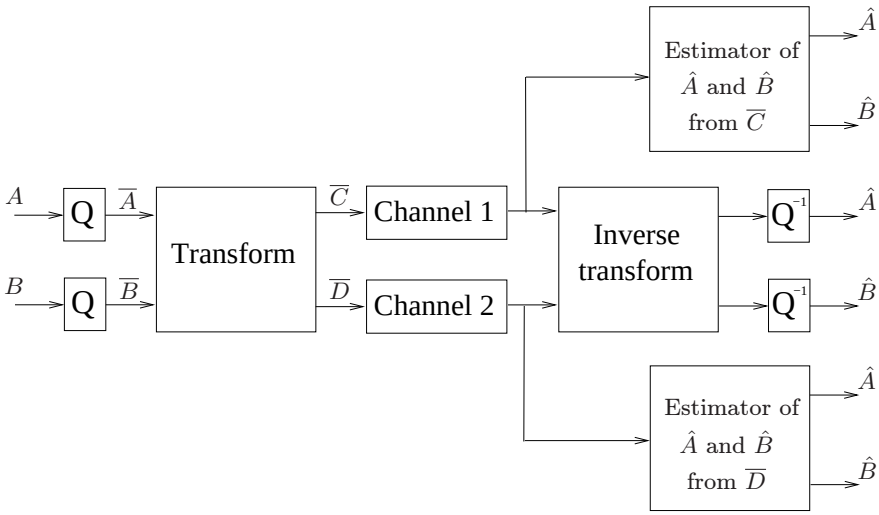


Figure 3.1. The basic structure of MDC using pairwise correlating transforms as presented in [50].

$$\begin{bmatrix} \overline{C} \\ \overline{D} \end{bmatrix} = \mathbf{T} \begin{bmatrix} \overline{A} \\ \overline{B} \end{bmatrix}, \tag{3.1}$$

where $\mathbf{T}$ is a 2×2 matrix. This transform is invertible so that

$$\begin{bmatrix} \overline{A} \\ \overline{B} \end{bmatrix} = \mathbf{T}^{-1} \begin{bmatrix} \overline{C} \\ \overline{D} \end{bmatrix}. \tag{3.2}$$

Once the data have been transformed $\overline{C}$ and $\overline{D}$ are transmitted over two different channels. If both the descriptors are received the inverse transform from (3.2) is used in order to produce $\hat{A}$ and $\hat{B}$. However, if one of the descriptors is lost, $\hat{A}$ and $\hat{B}$ can be estimated from the other descriptor. This comes from the fact the the transform matrix $\mathbf{T}$ is nonorthogonal and introduces redundancy in the transmitted data. For instance, if the receiver receives only the descriptor $\overline{C}$, $(\hat{A}, \hat{B})$ is estimated to $E[(A, B)|\overline{C}]$.

For the two descriptors case the transform matrix $\mathbf{T}$, optimized according to [50], can be written as

$$\mathbf{T} = \begin{bmatrix} \cos\theta/\sin 2\theta & \sin\theta/\sin 2\theta \\ -\cos\theta/\sin 2\theta & \sin\theta/\sin 2\theta \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{3.3}$$

where θ will control the amount of introduced redundancy.

The values $\overline{C}$ and $\overline{D}$ that are to be transmitted should be integers which is not necessarily the case in (3.1). Therefore the transform is implemented as follows (a, b, c and d are the values from (3.3) and $[\cdot]$ denotes rounding).

$$\overline{A} = \left[\frac{A}{A_{max}} q_A + 0.5\right], \quad \overline{B} = \left[\frac{B}{B_{max}} q_B + 0.5\right], \tag{3.4}$$

$$W = \overline{B} + \left[\frac{1+c}{d}\overline{A}\right], \tag{3.5}$$

$$\overline{D} = [dW] - \overline{A}, \tag{3.6}$$

$$\overline{C} = W - \left[\frac{1-b}{d}\overline{D}\right]. \tag{3.7}$$

It is assumed that $A \in [0, A_{max}]$ and $B \in [0, B_{max}]$. q_A and q_B are integers deciding how many quantization levels there are for A and B respectively. It is also assumed, for the extremes, that $\left[\frac{0}{A_{max}} q_A + 0.5\right]$ is rounded to 1 and $\left[\frac{A_{max}}{A_{max}} q_A + 0.5\right]$ is rounded to q_A.

Assuming that both descriptors are received in the decoder the corresponding inverse transform is performed as

$$W = \overline{C} + \left[\frac{1-b}{d}\overline{D}\right], \tag{3.8}$$

$$\overline{A} = [dW] - \overline{D}, \tag{3.9}$$

$$\overline{B} = W - \left[\frac{1+c}{d}\overline{A}\right], \tag{3.10}$$

$$\hat{A} = \frac{(\overline{A} - 0.5)A_{max}}{q_A}, \quad \hat{B} = \frac{(\overline{B} - 0.5)B_{max}}{q_B}. \tag{3.11}$$

As mentioned before, if one of the descriptors is lost $\hat{A}$ and $\hat{B}$ are, depending on which descriptor that was lost, estimated to $E[(A, B)|\overline{C}]$ or $E[(A, B)|\overline{D}]$.

Note here that the number of quantization levels for $\overline{A}$ and $\overline{B}$, q_A and q_B, will in general not equal the ones for $\overline{C}$ and $\overline{D}$, q_C and q_D. (q_A, q_B) are however mapped to (q_C, q_D) by a function φ according to

$$\begin{aligned} &\varphi : \mathbf{N}^2 \longrightarrow \mathbf{N}^2, \\ &\varphi(q_A, q_B) = (q_C, q_D). \end{aligned} \tag{3.12}$$

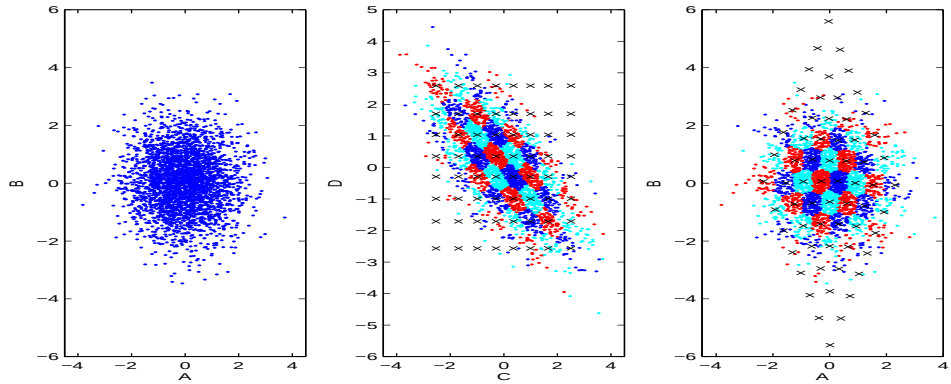


Figure 3.2. In the left plot the original set of data is shown. These values are first transformed and then quantized as shown in the middle plot. In the receiver the inverse transform is used as shown in the right plot. In this plot also the corresponding quantization cells are illustrated.

Hence, if we want to transmit $\overline{C}$ and $\overline{D}$ using, e.g., 3 bits each we need to find q_A and q_B so that $\varphi(q_A, q_B) = (2^3, 2^3)$.

From (3.4) it is seen that the described MDC system in (3.4)–(3.11) uses uniform quantization. The system could easily be improved by introducing two nonuniform scalar quantizers, one for the A-values and one for the B-values. This improved system is what will be used and considered further on in this chapter. This leads to modifications of (3.4) and hence also (3.11). Using the MSE as a distortion measure a codebook could be designed by using for instance the generalized Lloyd algorithm briefly explained in Section 3.3.

3.3 Improving the Quantization

In brief the algorithm in Section 3.2 can be summarized as

1. *Train encoder/decoder and quantize data.* The encoder uses two scalar quantizers in order to decrease the entropy of the data. This means that the data values are mapped onto a set of codevectors.

2. *Transform the quantized data.* Redundancy is introduced into the data by using (3.5)–(3.7).

3. *Transmit data.* The data is transmitted and packet or bit losses may occur, which means that some descriptors may be lost.

4. *Estimate lost data and do the inverse transform.* This is done using (3.8)–(3.10).

In this chapter we suggest to do this algorithm in a different order. Changing the order of Steps 1 and 2 would mean that the transformation is done directly and training and quantization is done on the transformed values. Naturally, also the order in the receiver has to be reversed appropriately.

Using MSE as the distortion measure a point in the data is quantized to the K:th codevector according to

$$\begin{aligned} K &= \arg\min_k \left(\begin{bmatrix} A \\ B \end{bmatrix} - \begin{bmatrix} \tilde{A}_k \\ \tilde{B}_k \end{bmatrix} \right)^T \left(\begin{bmatrix} A \\ B \end{bmatrix} - \begin{bmatrix} \tilde{A}_k \\ \tilde{B}_k \end{bmatrix} \right) \\ &= \arg\min_k \left(\begin{bmatrix} \Delta A_k \\ \Delta B_k \end{bmatrix}^T \begin{bmatrix} \Delta A_k \\ \Delta B_k \end{bmatrix} \right), \end{aligned} \tag{3.13}$$

where $\tilde{A}_k$ and $\tilde{B}_k$ are the coordinates of the different codewords. Using (3.2) this can also be written

$$\begin{aligned} K &= \arg\min_k \left(\mathbf{T}^{-1} \left(\begin{bmatrix} C \\ D \end{bmatrix} - \begin{bmatrix} \tilde{C}_k \\ \tilde{D}_k \end{bmatrix} \right) \right)^T \\ &\qquad \cdot \left(\mathbf{T}^{-1} \left(\begin{bmatrix} C \\ D \end{bmatrix} - \begin{bmatrix} \tilde{C}_k \\ \tilde{D}_k \end{bmatrix} \right) \right) \\ &= \arg\min_k \left(\mathbf{T}^{-1} \begin{bmatrix} \Delta C_k \\ \Delta D_k \end{bmatrix} \right)^T \left(\mathbf{T}^{-1} \begin{bmatrix} \Delta C_k \\ \Delta D_k \end{bmatrix} \right) \\ &= \arg\min_k \left(\begin{bmatrix} \Delta C_k \\ \Delta D_k \end{bmatrix}^T \mathbf{T}^{-1^T} \mathbf{T}^{-1} \begin{bmatrix} \Delta C_k \\ \Delta D_k \end{bmatrix} \right). \end{aligned} \tag{3.14}$$

According to the discussion in Section 3.2 there should be q_C quantization levels for C and q_D quantization levels for D. Introducing this restriction in (3.14) and using (3.3) gives

$$(I, J) = \arg\min_{i,j} (\Delta C_i^2 + 2\cos(2\theta)\Delta C_i \Delta D_j + \Delta D_j^2), \tag{3.15}$$

where $i \in \{1, 2, \ldots, q_C\}$ and $j \in \{1, 2, \ldots, q_D\}$. This equation will allow us to design a codebook for the transformed values instead of the original

data. The generalized Lloyd algorithm can be used for this purpose. This algorithm is briefly summarized below.

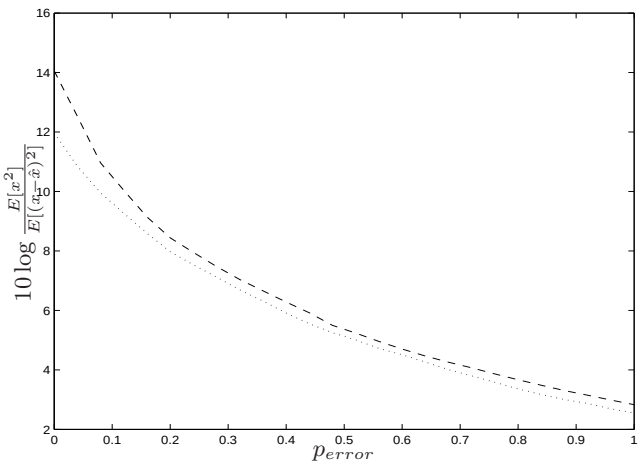


Figure 3.3. The dotted line shows the performance of the original system [50] and the dashed line shows that of the new system, in terms of signal-to-distortion ratio versus packet loss rate, p_{error}. $\overline{C}$ and $\overline{D}$ are transmitted using 3 bits each and $\theta = \frac{\pi}{5}$.

1. Define initial codebook.
2. Quantize each data point to that codeword that minimizes the contribution to the distortion.
3. For each codeword (if it is possible), find a new optimal codeword for all the values that have been quantized to this particular codeword and update the codebook.
4. Until the algorithm converges go to Step 2.

For Step 2, (3.15) is used to quantize the data. In Step 3 we want to find an optimal codeword for those values that have been quantized to a particular codeword. Calculating the partial derivative of the total distortion as

$$\frac{\partial}{\partial \tilde{C}_I} \sum_{(C,D)} (\Delta C_i^2 + 2\cos(2\theta)\Delta C_i \Delta D_j + \Delta D_j^2) \tag{3.16}$$

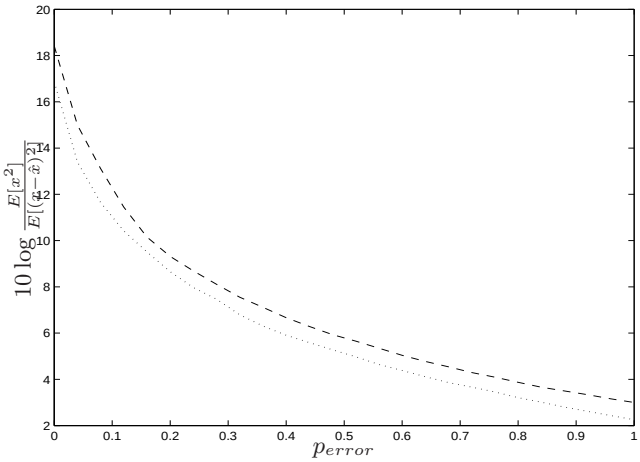


Figure 3.4. The dotted line shows the performance of the original system [50] and the dashed line shows that of the new system, in terms of signal-to-distortion ratio versus packet loss rate, p_{error}. $\overline{C}$ and $\overline{D}$ are transmitted using 4 bits each and $\theta = \frac{\pi}{5}$.

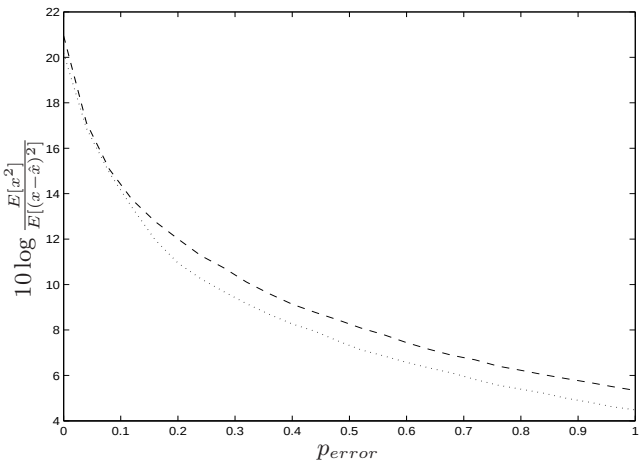


Figure 3.5. The dotted line shows the performance of the original system [50] and the dashed line shows that of the new system, in terms of signal-to-distortion ratio versus packet loss rate, p_{error}. $\overline{C}$ and $\overline{D}$ are transmitted using 8 bits each and $\theta = \frac{\pi}{5}$.

and minimizing by setting (3.16) equal to zero will give an equation for updating the codevectors, namely

$$\tilde{C}_I = \frac{1}{N_I} \sum_{\forall (C,D): Q(C,D)=(\tilde{C}_I, \tilde{D}_j)} (C + \cos(2\theta)\Delta D_j). \tag{3.17}$$

The sum is taken over all those points (C, D) which will be quantized to $(\tilde{C}_I, \tilde{D}_j)$ for a given I and an arbitrary j. N_I is the number of points within this set. In a similar manner we get

$$\tilde{D}_J = \frac{1}{N_J} \sum_{\forall (C,D): Q(C,D)=(\tilde{C}_i, \tilde{D}_J)} (D + \cos(2\theta)\Delta C_i) \tag{3.18}$$

and this is done for $I = 1, 2, \ldots, q_C$ and $J = 1, 2, \ldots, q_D$. Once the codebook has been generated the encoder and decoder are ready to use. The data to be transmitted is then transformed by the matrix $\mathbf{T}$, quantized and transmitted. In the decoder the reverse procedure is done. This is illustrated in Figure 3.2.

3.4 Simulation Results

In order to compare the system explained in Section 3.2 and [50] with the new system introduced in Section 3.3 these were implemented and simulated. Uncorrelated zero mean Gaussian data was generated and used to train the encoders/decoders and then to simulate the systems. In the simulations presented here the source data A and B have equal variances. Similar results have however been obtained also for the case of non-equal variances. As mentioned in Section 3.1 it is assumed that only one descriptor can be lost at a time and that the receiver knows when a descriptor is lost. The angle for the transform matrix $\mathbf{T}$ used in the simulations was $\theta = \frac{\pi}{5}$. The result is presented in Figures 3.3–3.5. p_{error} show the probability that one of the descriptors is lost and the y-axis shows the signal–to–distortion ratio, defined as $10 \log \frac{E[x^2]}{E[(x-\hat{x})^2]}$, where x is the data signal and $\hat{x}$ is the reconstructed signal. In Figure 3.3 both $\overline{C}$ and $\overline{D}$ were transmitted using 3 bits each which gives $q_C = q_D = 2^3$. In order to accomplish this (q_A, q_B) had to be identified so that $\varphi(q_A, q_B) = (2^3, 2^3)$. This was found to be true for $q_A = 5$ and $q_B = 7$. Similar results are shown in Figures 3.4 and 3.5 when using 4 and 8 bits.

As can be observed in Figures 3.3–3.5 the new system outperforms the original system for all investigated values of p_{error}. In the case of 3 bits

per description, as shown in Figure 3.3, the advantage of the new scheme is more noticeable at low packet loss rates. In particular we see that as $p_{error} \rightarrow 0$ the new system outperforms the original scheme by about 2 dB. When using 4 bits per description, as in Figure 3.4, we notice that the gain of the new approach is more-or-less constant over the range of different packet loss rates. Finally, studying Figure 3.5, we can observe that in the case of 8 bits per description the situation has changed and the gain is now more prominent at high packet error rates. In summary we see that in all cases considered there is a constant gain at medium to high packet loss rates and this gain increases with the transmission rate of the system, while at low packet loss rates there is an additional gain at low rates (as in Figure 3.3) and hardly no gain at high rates (as in Figure 3.5). One possible explanation for this behavior is that the new approach in particular improves the performance at low transmission and packet loss rates due to the improved optimization of the individual quantizers. At high loss rates this gain is less pronounced, since when packet losses occur the redundancy introduced by the linear transform has an equal or higher influence on the total performance than has the performance of the individual quantizers.

3.5 Conclusions

A new MDC method has been introduced. The method is developed from an extended version of the MDC using pairwise correlating transforms described in [50]. Using the original method the data is quantized and then transformed by a matrix operator in order to increase the redundancy between descriptors. In the new suggested method the data is first transformed and then quantized. In Section 3.3 it is shown that this transform 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. In section 3.4 simulations were done that shows that the new method performs better than the original one when smaller amounts of redundancy are introduced into the transmitted data. For the simulations conducted in Section 3.4, using $\theta = \frac{\pi}{5}$, the new method gave 2 dB 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.

Chapter 4

Image Coder

This chapter presents a simple, yet effective, image coder that is used later in this thesis for evaluating the proposed joint source and channel coding methods in a more realistic system. It should be stressed that the intention is *not* to create a top of the notch, best ever image coder, in terms of compression. That would require schemes that are overly complex for our purpose. Instead, the structure of the image coder is intentionally kept as simple as possible. The most important reason is that it should be able to handle severe channel conditions without breaking down completely. As an example, the image coder does not use entropy coding. A choice that surely degrades the performance in terms of pure compression. The reason is simple; any error introduced in an entropy-coded bit-stream is likely to destroy all the following data due to error propagation.

The remainder of this chapter is organized as follows. First the basic structure of the image coder is described. The two main components of the image coder, subband coding and vector quantization, are discussed in detail. Next follows a discussion on how to model the statistics of vectors from the image subbands. This discussion includes a description of Gaussian mixture models, and the expectation maximization algorithm. Finally, some examples of images are included that have been encoded using our newly devised image coder.

4.1 Image Coder Structure

The image coder consists of two main parts—an image transform, followed by vector quantization. The image transform serves to remove statistical redundancy, or correlation, from the image. Image transforms come in several different flavors, and for this particular image coder, a subband transform is used.

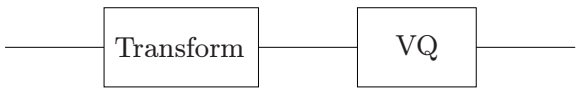


Figure 4.1. Basic structure of the image coder

After the image transform, the transform coefficients have to be quantized to get a discrete representation of the image that can be encoded into a stream of bits. For this purpose, the image coder uses vector quantization. The choice of vector quantization instead of scalar quantization, is partly to compensate for some of the performance loss we get by not using entropy coding, and partly to allow the image coder to use the robust quantization framework discussed in Section 2.2.

4.1.1 Subband Image Transform

The transform used in the image coder is a 2-dimensional subband transform. A subband transform is obtained by splitting the source into different representations, subbands, corresponding to different spectral content of the source. Such splitting into different representations can be implemented by using *filter banks*, as is explained in the remainder of this section.

The basic building block of the subband transform is a 2-channel filter bank, depicted in Figure 4.2. Two different filter banks are needed, one for analysis and one for synthesis. The analysis filter bank acts as our forward transform, and performs the actual splitting of the input into two different parts. Figure 4.3 shows a schematic picture of the frequency response of the analysis filters. Since each filter cuts the bandwidth of the input signal in half, each subband can be decimated by a factor 2 without any loss of information. We refer to the output as *transform coefficients*. Since the output from the filters is subsampled, the number of transform coefficients

is equal to the number of samples in the input signal. In the literature such filter banks are often called *critically sampled* filter banks.

The synthesis filter bank performs the reverse operation and acts as our inverse transform. Obviously we want the output from the inverse transform to be as close to identical to the input of the forward transform as possible. Without going into the details, it turns out that it is indeed possible to construct filters H_0, H_1, G_0 and G_1 such that the output is *exactly* equal to the input. Such filter banks are said to have the *perfect reconstruction* property. There are two conditions that have to be satisfied in order to achieve perfect reconstruction:

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0 \tag{4.1}$$

and

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2. \tag{4.2}$$

There exists a variety of filters in the literature that satisfy the two above constraints. Ideally, one would like to have finite length, linear phase filters that give an orthogonal transform. Unfortunately, there are no filters that satisfy all three wishes, except for the trivial case of Haar filters. Orthogonal transforms are attractive because they offer a simple way to analyze and predict performance directly in the transform domain, as described in Section 1.1.4. More specifically, if the coefficients of an orthogonal transform are approximated by $y_k \approx \hat{y}_k$, then because of the energy conserving property of orthogonal transforms (1.17) the total error in a mean squared sense is equal in the transform domain and the image domain, i.e. $\sum_n |x_n - \hat{x}_n|^2 = \sum_k |y_k - \hat{y}_k|^2$. This makes it easy to analyze the effects of e.g. quantization of transform coefficients.

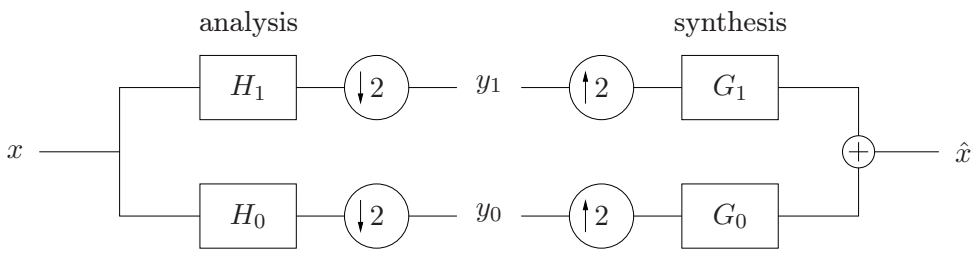


Figure 4.2. 2-channel filter bank

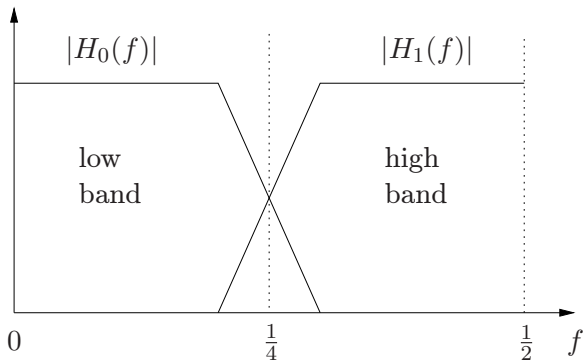


Figure 4.3. Splitting of spectrum into two bands

So far the discussion has mainly been about one-dimensional transforms. But images are two-dimensional by nature, so we need to extend the results to obtain two-dimensional transforms. For this, a separable approach is taken; the one-dimensional transform is applied first to each row of image data, and then to each column of the horizontally transformed data as shown in Figure 4.4. The actual ordering does not matter, only that the transform is applied once along each dimension. This is equivalent to using a separable 2-dimensional filter together with 2-dimensional separable subsampling. Non-separable filtering is also possible, and offers more flexibility, but is also more complex to analyze and implement.

The result of the transform in Figure 4.4 is four different subbands. The subbands are named LL, HL, LH, and HH to indicate which filtering operations have been performed. Thus, the HL-band for has been highpass filtered in the horizontal direction and lowpass filtered in the vertical direction. The naming convention is analogous for the other subbands.

Since the number of transform coefficients is the same as the number of input samples in the image, it is possible to illustrate all four subbands at the same time by putting them next to each other as in Figure 4.5. The high pass subbands all have a mean value of 0, which is represented by gray in Figure 4.5. Bright and dark pixels correspond to positive and negative transform coefficients respectively. The figure shows clearly that most coefficients in the highest subbands are close to zero. This is a motivating factor in using a subband transform for compression purposes, since these coefficients can be efficiently quantized using a low number of bits. One

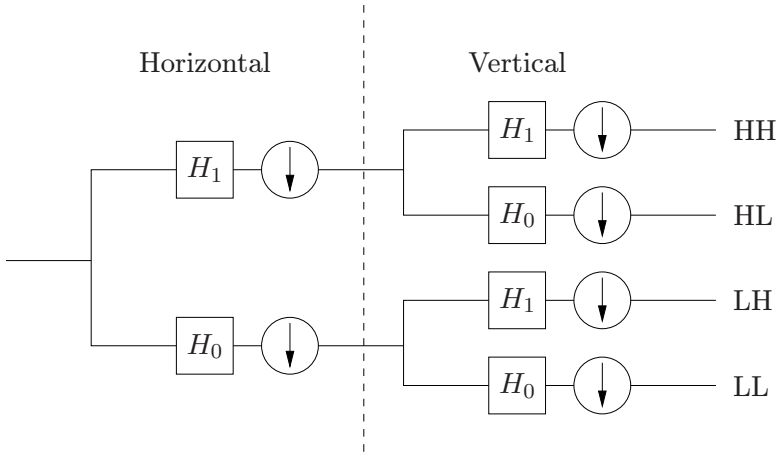


Figure 4.4. Implementing separable subband transform. Only analysis filter bank shown

observation that should be made is that the lowest subband is nothing but a smaller version of the original image. It is intuitive that we can increase the compression performance by applying the subband transform again on the LL subband. Figure 4.6 shows a block diagram of this approach, and the resulting transform coefficients are displayed as an image in Figure 4.7.

The transform used in the image coder uses four recursive splits, giving a total of 13 subbands. The resulting transform is shown in Figure 4.8 with a numbering of the image subbands for future reference.

4.1.2 Vector Quantizer

Vector quantization of image subband coefficients can be done in several different ways. A good overview can be found in the tutorial paper [8].

Ideally, the image transform would remove all of the statistical redundancy in the image, resulting in transform coefficients that are completely uncorrelated. This is not the case for most natural images. Coefficients in the subband corresponding to the lowest frequencies are obviously still correlated, as seen in Figures 4.5, 4.7, but there is also residual redundancy between the coefficients of other subbands as well. Take the HL subband for example. It has been highpass filtered in the horizontal direction and lowpass filtered in the vertical direction. In this subband, vertical edges in

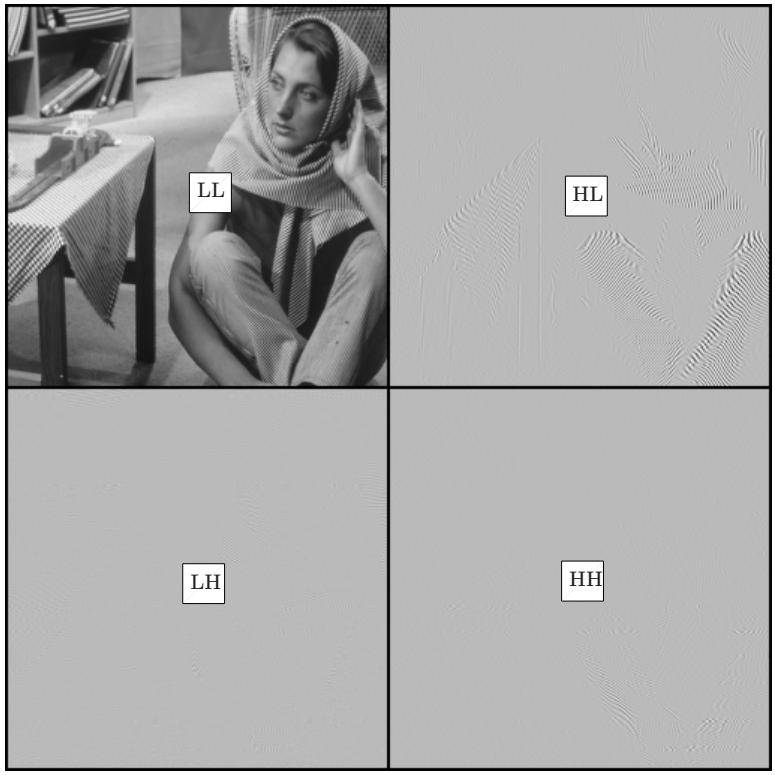


Figure 4.5. Illustration of the effects of a subband transform on the Barbara image. Subbands are placed next to each other to keep the original dimensions with lowest frequency to the top left.

the image show up as vertical lines. This is because the highpass filtering in the horizontal direction leaves the finest details in the horizontal direction, e.g. the sharp transition of a vertical edge, while the vertical direction is of lowpass character. This suggests that there is more correlation left in the vertical direction in the HL subband. By similar reasoning, the LH subband contains more correlation in the horizontal direction. Since a vector quantizer can take advantage of correlation between components, it seems like a good idea to select vectors such that neighboring coefficients with correlation end up in the same vector.

Figure 4.10 illustrates how vectors are selected in the image coder. Each subband uses its own vector quantizer, i.e. vectors are formed from coefficients within the subband only. Coefficients are selected in the direction that

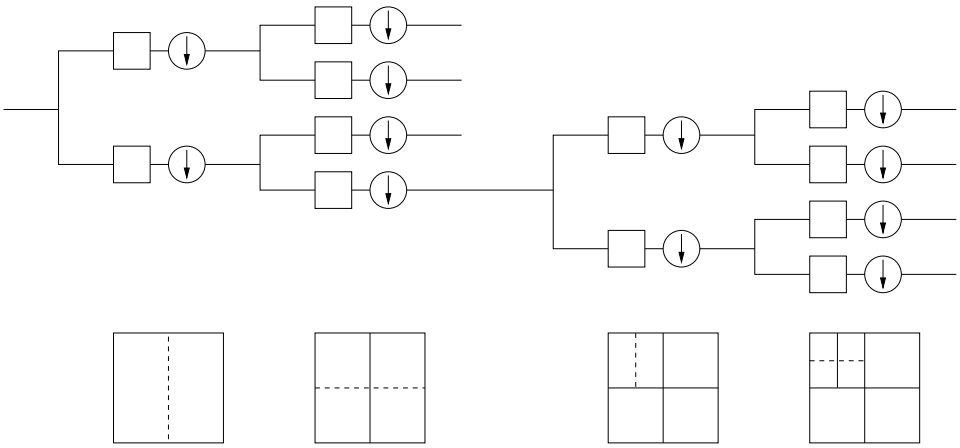


Figure 4.6. Recursive splitting of the low frequency subband. The dashed line indicates the split taking place in each step of the transform.

is most probable to contain residual redundancy. The highest frequency subbands require less rate in the encoding and therefore use 4-dimensional VQ's. Lower frequency subbands that require more rate use 2-dimensional VQ's in order to keep the computational complexity and storage requirements down. The vector size selected for each subband is given in Table 4.1.

4.2 Probability Distribution of Transform Coefficients

Before we are ready to design the vector quantizers, we need to know the probability distribution of the transform coefficients. The standard way to proceed, is to simply take a large number of representative images and use the transformed data as samples of the source, and use them directly as training data. This approach is simple, but has a number of drawbacks. The most important drawback is that the number of coefficients in the lowest subbands is small. At the same time, those subbands have coefficients with larger variance, and require more rate in the encoding. Since a higher rate means a larger codebook to train, we would actually want a large number of samples in the lowest frequency subbands. This means that a very large number of images would have to be used, and even then, the accuracy of the training might leave more to wish.

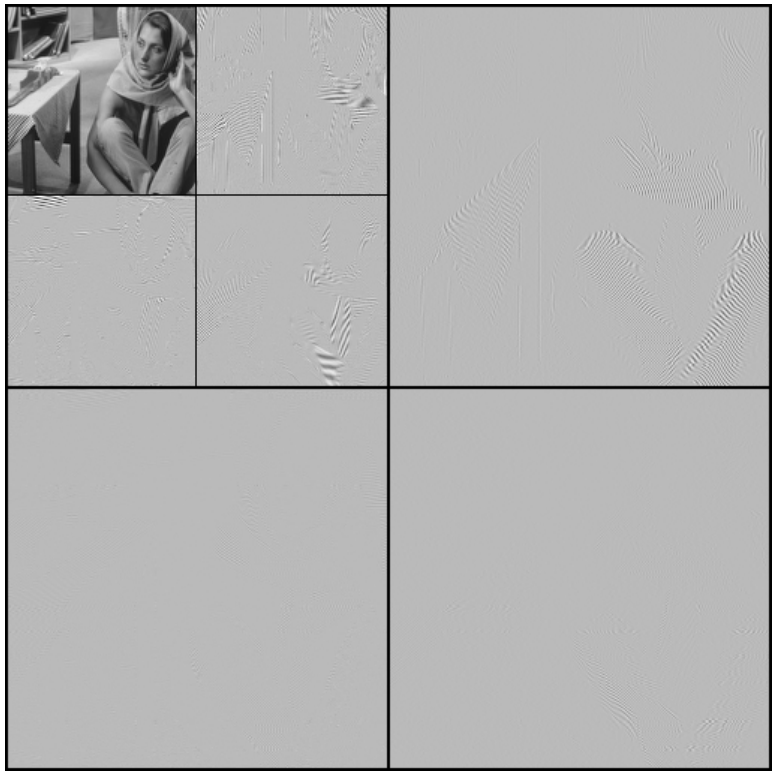


Figure 4.7. Illustration of wavelet transform

This section describes an alternative approach, that is based on modeling the probability distribution of the transform coefficients. Having a good model means that training data can be synthesized to the extent that is needed in order to achieve good VQ training. This approach also has the extra advantage that over-fitting a vector quantizer to a small set of training data can be avoided to some extent, a fact that has been observed by others and was reported in [22].

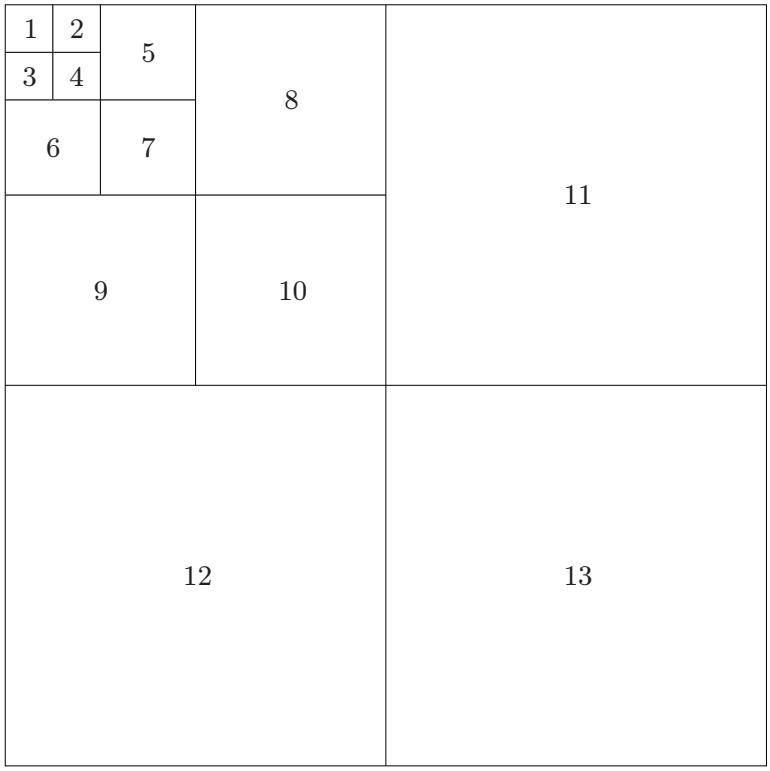


Figure 4.8. Image subbands in the transform used in the image coder

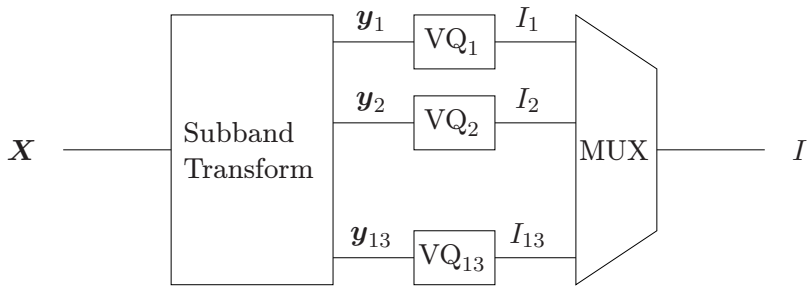


Figure 4.9. Schematic figure of the image coder illustrating that each subband uses a separate VQ.

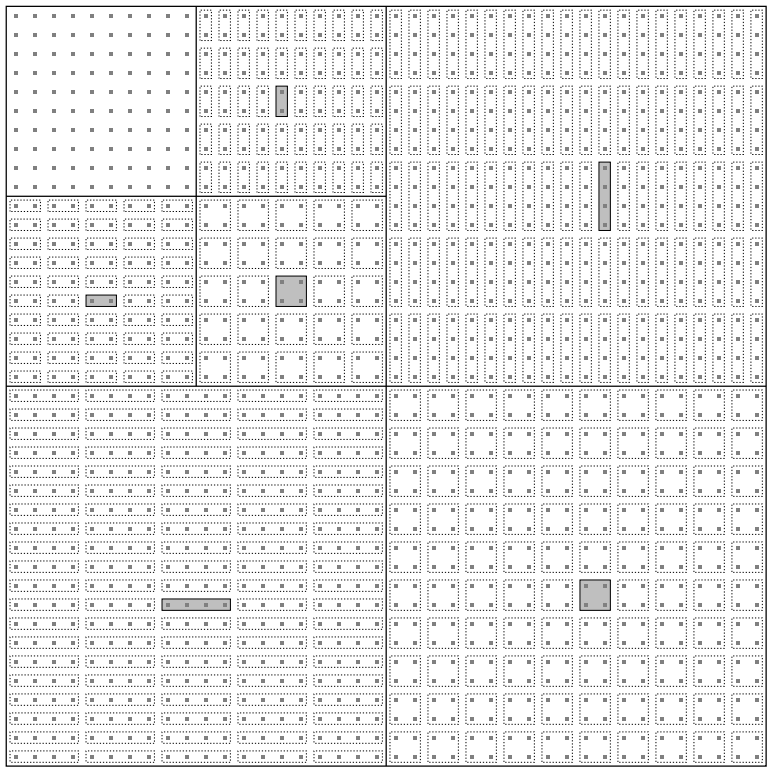


Figure 4.10. Illustration of how vectors are selected from different subbands in order to utilize residual redundancy. The small dots correspond to transform coefficients and vectors are indicated by dotted boxes around the coefficients. One vector in each subband has been marked gray to better indicate the shape of the vector.

Subband number	1	2	3	4	5	6	7	8	9	10	11	12	13
Vector size	1	2	2	2	4	4	4	4	4	4	4	4	4

Table 4.1. Vector sizes used for each subband in the image coder

4.2.1 Utilizing Self-Similarity of the Transform

Since the transform is recursive, there is a certain amount of similarity between the subbands. In Section 4.1.1, the recursive division of the low frequency subbands was motivated by the fact that the LL-band in each stage of the transform is essentially a decimated version of the image. Therefore, it is reasonable to assume that the probability distributions of subbands at different stages of the recursion are similar. For instance, the probability distribution of the horizontal detail subbands 2, 5, 8 and 11 in Figure 4.8 should be similar in some way.

To elaborate further on this similarity, consider the case when orthogonal, or close to orthogonal filters are used in the filter bank. Then the transform has the conservation-of-energy property. In the first stage of the transform, the image is split in four parts. Since the energy in the detail subbands is close to zero, and the transform conserves energy, almost all energy will be found in the LL subband. This means that since the number of coefficients in the LL-band is four times smaller than the number of pixels in the image, the average energy of each coefficient must be approximately four times larger than the average energy of the image pixels. In other words, the amplitude of the LL-band coefficients is about twice the amplitude of the image pixels.

From the above argument, we make the assumption that the probability distribution of similar subbands, e.g. all horizontal detail subbands, have *the same shape*, but is *scaled by a factor two* in each step of the transform. In other words we assume that if $f_y^{(11)}(\boldsymbol{y})$ denotes the pdf of a vector in the subband 11, then $f_y^{(8)}(\boldsymbol{y}) = \frac{1}{2} f_y^{(11)}(\frac{\boldsymbol{y}}{2})$ is the pdf of a vector of the same size in the subband 8, using the subband numbering of Figure 4.8.

It seems that based on the assumptions made above, it would be sufficient to have four models of subband coefficients, one for the lowpass band, and one each for the vertical, horizontal and diagonal detail bands. It turns out that extending the reasoning a bit further allows us to use only three models. Consider what happens if an image is rotated 90 degrees clockwise and then flipped in the horizontal direction, i.e. the same as representing the image as a matrix and taking the transpose. The result is the same as if shooting a mirror image with a camera held in portrait mode. In other words, this is an equally valid natural image. The point of performing this operation is that the role of the horizontal detail subbands and the vertical detail subbands are swapped, i.e. the result is the same as if the subbands are arranged as an image like in Figure 4.7 and then transposed. This means that it is fair to

assume that vectors from horizontal and vertical detail subbands have the same pdfs, given that vectors are selected as in Figure 4.10.

To summarize, three different models are needed, one for the lowpass subband, one for horizontal/vertical detail subbands, and one for diagonal detail subbands.

4.2.2 Gaussian Mixture Models

Gaussian mixture (GM) models are models of probability density functions with a well known ability to model arbitrary pdfs. They are well suited for use as underlying models of probability density functions in the design of vector quantizers, a topic that was investigated in [22].

As the name implies, a GM density consists of a mixture of Gaussian density functions, where the word *mixture* should be read as *weighted sum*. Assume that we want to model the pdf $f_{\boldsymbol{X}}(\boldsymbol{x})$ of a random variable $\boldsymbol{X}$ with a Gaussian mixture model. Then

$$f_{\boldsymbol{X}}(\boldsymbol{x}) \approx f_M(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{M} \rho_i f_i(\boldsymbol{x};\boldsymbol{\theta}), \tag{4.3}$$

where $f_i(\boldsymbol{x};\boldsymbol{\theta})$ is a multivariate Gaussian density

$$f_i(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{C}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{C}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}$$

with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{C}_i$. The component weights, $\rho_i > 0,\ i \in \{1 \ldots M\}$, sum up to unity, $\sum_{i=1}^{M} \rho_i = 1$, in order to make $f_M(\boldsymbol{x};\boldsymbol{\Theta})$ a true pdf, in the sense that it should integrate to unity. Note that the GM density is completely specified by its parameters, defined by the set $\boldsymbol{\Theta} = \{\rho_1, \ldots, \rho_M, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$, where $\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \boldsymbol{C}_i\}$.

4.2.3 Expectation Maximization Algorithm

In order to fit a GM model to a set of training vectors, the parameter set $\boldsymbol{\Theta}$ has to be estimated. Several different methods exist, and the one used most often is known as the expectation maximization (EM) algorithm. The EM algorithm is an iterative method that is widely used for maximum likelihood (ML) estimation in situations where closed form analytical solutions are hard to find. In the case of GM modeling, assume that there is a set of

N_x vectors, $\{\boldsymbol{x}_n\}_{n=1}^{N_x}$, from the density that should be modeled. Then the log-likelihood criterion to maximize is defined by

$$L(\boldsymbol{\Theta}) = \log \prod_{n=1}^{N_x} f_M(\boldsymbol{x}_n; \boldsymbol{\Theta}) = \sum_{n=1}^{N_x} \log f_M(\boldsymbol{x}_n; \boldsymbol{\Theta}). \tag{4.4}$$

Since $L(\boldsymbol{\Theta})$ contains a sum of logarithms, it is difficult to find an analytical solution to the maximization problem. Instead, the EM-algorithm solves the problem iteratively, by approximating the solution based on the result of the previous iteration.

The general EM algorithm is based on expanding the data set in a way such that

$$\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M) = \boldsymbol{g}(\boldsymbol{y}).$$

In other words, $\boldsymbol{g}$ is a many-to-one transformation that maps the *complete data* $\boldsymbol{y}$ into the *incomplete data* $\boldsymbol{x}$. $\boldsymbol{y}$ contains all the information about $\boldsymbol{x}$, but not vice versa. Maximizing $\log f(\boldsymbol{x}; \boldsymbol{\Theta})$ is difficult, so $\log f(\boldsymbol{y}; \boldsymbol{\Theta})$ is maximized instead. Since $\boldsymbol{y}$ is unavailable, the log-likelihood function is replaced by its conditional expectation

$$E\left[\log f(\boldsymbol{y}; \boldsymbol{\Theta}) | \boldsymbol{x}\right] = \int f(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\Theta}) \log f(\boldsymbol{y}; \boldsymbol{\Theta}) d\boldsymbol{y}.$$

Since $\boldsymbol{\Theta}$ has to be known in order to determine $f(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\Theta})$, the estimate from the previous iteration is used. Let $\boldsymbol{\Theta}^{(k)}$ denote the estimate after k iterations, then the EM-algorithm consists of the following iterative steps:

Expectation: Determine the average log-likelihood function

$$U(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(k)}) = \int f(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\Theta}^{(k)}) \log f(\boldsymbol{y}; \boldsymbol{\Theta}) d\boldsymbol{y}. \tag{4.5}$$

Maximization: Find

$$\boldsymbol{\Theta}^{(k+1)} = \arg\max_{\boldsymbol{\Theta}} U(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(k)}). \tag{4.6}$$

Closed form solutions of the update equation exist for many problems. In the case of Gaussian mixture estimation, the complete data is chosen such that $f(\boldsymbol{y}_i; \boldsymbol{\Theta}) = \rho_i f(\boldsymbol{x}; \boldsymbol{\Theta}_i)$, and the update equations that solve (4.6) are given by

$$\rho_i^{(k+1)} = \frac{1}{N_x} \sum_{n=1}^{N_x} \nu_i^{(k)}(n) \tag{4.7}$$

$$\boldsymbol{\mu}_i^{(k+1)} = \frac{\sum_{n=1}^{N_x} \nu_i^{(k)}(n) \boldsymbol{x}_n}{\sum_{n=1}^{N_x} \nu_i^{(k)}(n)} \tag{4.8}$$

$$\boldsymbol{C}_i^{(k+1)} = \frac{\sum_{n=1}^{N_x} \nu_i^{(k)}(n) (\boldsymbol{x}_n - \mu_i^{(k+1)})(\boldsymbol{x}_n - \mu_i^{(k+1)})^T}{\sum_{n=1}^{N_x} \nu_i^{(k)}(n)} \tag{4.9}$$

where $\nu_i^{(k)}(n)$ denotes the posterior probabilities $f(\boldsymbol{y}_i|\boldsymbol{x}_n; \boldsymbol{\Theta}^{(k)})$, defined by

$$\nu_i^{(k)}(n) = \frac{\rho_i^{(k)} f_i(\boldsymbol{x}_n; \boldsymbol{\theta}_i^{(k)})}{\sum_{j=1}^{M} \rho_j^{(k)} f_j(\boldsymbol{x}_n; \boldsymbol{\theta}_j^{(k)})}. \tag{4.10}$$

The EM-algorithm has the attractive property that each iteration increases the likelihood function, i.e. $L(\boldsymbol{\Theta}^{(k+1)}) \geq L(\boldsymbol{\Theta}^{(k)})$. This means that the algorithm is guaranteed to converge, at least to some local optimum.

4.3 Image Coder Summary

This section summarizes the different aspects of the image coder and gives a brief description of how to put the pieces together.

The first part of the image coder is a subband transform, described in Section 4.1.1. It is implemented as a filter bank, using the 32-tap Johnston QMF filters [23]. Four splits are used in the transform which gives 13 different subbands. The output from the filter bank is formed into vectors as illustrated in Figure 4.10. The vectors are then quantized using separate vector quantizers for each subband. The vector dimension for each subband is given in Table 4.1.

The design of the vector quantizers is based on Gaussian mixture models of the subband vector probability density functions. When designing the models, a value of $M = 4$ was used in (4.3), i.e. four components were used in each GM density. The design procedure is described in Table 4.2.

The result of the procedure in Table 4.2 is a set of vector quantizers at different rates for each subband vector pdf model. By generating new data

Table 4.2. Steps in the VQ design.

1. Collect a database of images for use as empirical distributions
2. Apply the image transform to each image.
3. Normalize each subband so the values are in the same range for all subbands, i.e. divide subbands 11–13 by 2, subbands 8–10 by 4, subbands 5–7 by 8, and subbands 1–4 by 16.
4. Collect vectors from subbands 2, 3, 5, 6, 8, 9, 11 and 12 and use as training database for designing the model corresponding to horizontal/vertical detail subbands.
5. Use the training database with the EM-algorithm to obtain the GM model for horizontal/vertical detail subband vectors.
6. Repeat with subbands 4, 7, 10 and 13 to get the GM model for the diagonal detail subband vectors.
7. Use the obtained models to generate training data for VQ training.
8. Train vector quantizers for different rates using the generated training data.

and quantizing it using the newly designed VQ's, an empirical distortion–rate function can be generated. The distortion–rate function has to be scaled for each subband, since the VQ-design was based on normalized vectors. The rate should also be scaled to the unit "bits per pixel in the original image". When empirical rate–distortion curves have been found for all subbands, bit allocation can be done as described in Chapter 1, Section 1.1.2, with the difference that the true distortion–rate functions are replaced by empirical ones. An example of a bit allocation for a total rate of 0.5 bits/pixel is given in Table 4.3. Note that the highest frequency detail subbands are not encoded at all.

Subband number	1	2	3	4	5	6	7	8	9	10	11	12	13
VQ rate	8	10	10	9	10	10	8	8	8	4	0	0	0

Table 4.3. Bit allocation for image coder at 0.5 bits/pixel

4.4 Image Examples

This section presents some image results when using the proposed image coder. Four different images were encoded at a rate of 0.5 bits/pixel using the bit allocation in Table 4.3. The result is presented in Figures 4.11–4.14. The (a) parts show the original images and the (b) parts show the encoded images. Note that throwing away the highest frequency subbands has resulted in some loss of detail. This is particularly noticeable in Figure 4.12, which contains lots of fine details.

(a)

(b)

Figure 4.11. (a) Original. (b) Encoded at 0.5 bits/pixel

(a)

(b)

Figure 4.12. (a) Original. (b) Encoded at 0.5 bits/pixel

(a)

(b)

Figure 4.13. (a) Original. (b) Encoded at 0.5 bits/pixel

(a)

(b)

Figure 4.14. (a) Original. (b) Encoded at 0.5 bits/pixel

Chapter 5

Robust Quantization for Channels with Both Bit Errors and Erasures

5.1 Introduction

For packet data networks where parts of the overall transmission are over wireless links, phenomena occur that are not present in traditional wired networks (over optical fiber and/or cable links). In particular, it is not a reasonable assumption to neglect bit-errors over wireless channels. In addition to bit-errors, source coder robustness relates to its sensitivity to packet-loss, which may occur in the network due to overload (or, sometimes, over wireless paths due to detected bit-errors in packets that are then declared lost).

The source–channel separation theorem, discussed in Chapter 2, Section 2.1, states that there is no loss in treating source and channel coding as two separate problems. As discussed in Chapter 2, the separation can however be made without loss only in the limit of infinite delay and coding complexity; a fact that has been frequently pointed out to motivate the use of *combined* source–channel codes. Many current source coders employ only error concealment to make the coder robust, while we emphasize the use of combined source–channel coding, with a focus on techniques based on channel optimized vector quantization.

The concept of COVQ originates in [4, 11, 12, 27, 53], and we introduced the basics in Chapter 2, Section 2.2.1. The use of COVQ has been proposed in many different contexts. However, even if much has already been said on the subject, most of the previous works have been focused towards channels

with bit-errors, such as the binary symmetric channel. In contrast, the main contributions of the present chapter are an extension to channels with *both* bit-errors *and* bit-erasures, and a demonstration of how the new COVQ technique can be implemented to enhance the performance of a subband image coder. This image coder was described in Chapter 4. Previous related work on error-robust image coding includes [6, 30, 36, 46].

5.2 The Binary Symmetric Erasure Channel

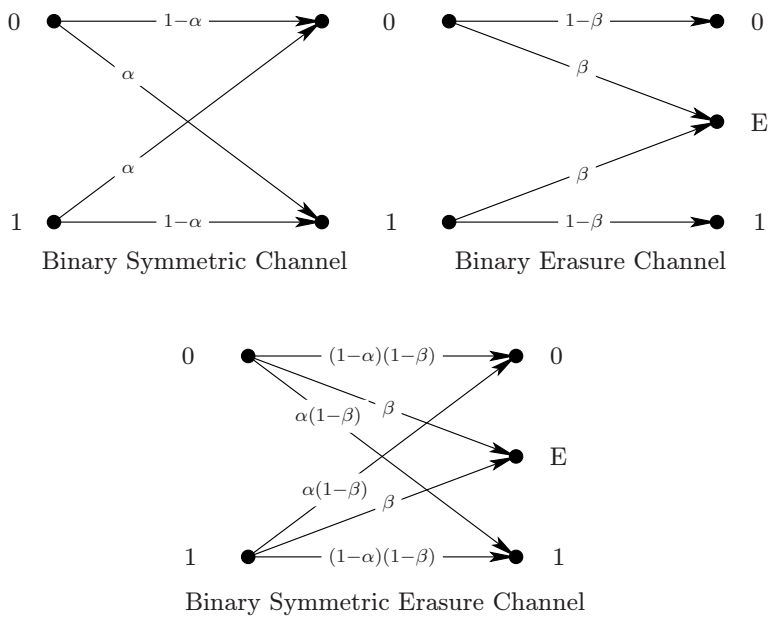


Figure 5.1. Three special cases of the BSEC

We consider channels where both erasures and bit-errors occur. Such channels can arise, e.g., when information is communicated over several consecutive channels with different properties. Consider for instance the case where a packet-switched network is used for long distance communication, but local access is made through a wireless link. The network may fail to deliver a packet on time, causing packet erasures. The wireless link, on the other hand, is prone to introduce bit-errors.

A wireless link, as defined by its physical properties, modulation, non-perfect channel coding, etc., can be replaced by an equivalent discrete channel. Often this discrete channel can be modeled as a *binary symmetric channel* (BSC) with a transition probability α, corresponding to the bit-error rate (BER) of the channel. In packet loss channels, erasures come in groups of many bits. We assume however, that bit-level interleaving, using a pseudo-random spreading sequence, is implemented in such a way that all bits in a *symbol* are transmitted in different packets. This way, from a symbol perspective, independent packet losses are turned into independent bit-erasures. The resulting channel is the *binary erasure channel*, defined by the erasure probability β which will be referred to as the bit-loss rate (BLR). Note that in our case the BLR is equal to the packet loss rate (PLR).

If the binary symmetric channel and the binary erasure channel are concatenated, the *binary symmetric erasure channel* (BSEC), depicted in Figure 5.1, is obtained. For the BSEC, bits are complemented with probability α and lost with probability β. Thus, the probability of a correctly received bit is $(1-\alpha)(1-\beta)$, the probability of a complemented bit is $\alpha(1-\beta)$ and the probability of an erasure is β. Note that both the binary symmetric channel and the erasure channel are obtained as special cases of the BSEC.

5.3 Channel Optimized VQ for the BSEC

For ease of reference, we repeat here some basic results about COVQ and COVQ design (see also Chapter 2, Section 2.2.1).

In general, a VQ or COVQ is defined by two basic operations, the encoder and decoder. The *encoder*, $\varepsilon(\cdot)$, transforms a source vector, $\boldsymbol{X} \in \mathbb{R}^k$, into a quantization index, $I = \varepsilon(\boldsymbol{X}), \quad I \in \mathcal{I}_M = \{0, 1, \ldots, M-1\}$. The encoder operation is defined by a partitioning, $\mathcal{P} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{M-1}\}$, of $\mathbb{R}^k$ such that $\varepsilon(\boldsymbol{x}) = i$, iff $\boldsymbol{x} \in \mathcal{S}_i$. The *decoder*, $\delta(\cdot)$, is a mapping from a finite set of integers to an associated set of vectors, $\boldsymbol{Y} = \delta(J), \quad J \in \{0, 1, \ldots, N-1\}$. The set of reconstruction vectors, $\mathcal{C} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N-1}\}, \quad \boldsymbol{y}_j \in \mathbb{R}^k$, is called the *decoder codebook*.

Suppose that the index $I = \varepsilon(\boldsymbol{X})$ is sent over a noisy channel, and that J is observed at the receiver. Assume also that the distortion measure $d(\boldsymbol{x}, \boldsymbol{y})$ associated with mapping an input vector, $\boldsymbol{x}$, into an output vector, $\boldsymbol{y}$, is given by the squared Euclidean distance, i.e.

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2 . \tag{5.1}$$

Then necessary conditions for minimizing the expected distortion,

$$D(\mathcal{P},\mathcal{C}) = E\left[d(\boldsymbol{X},\boldsymbol{Y})\right] \tag{5.2}$$

are

$$\mathcal{S}_i = \left\{ \boldsymbol{x} : \sum_{j=0}^{N-1} P(j|i) \left\|\boldsymbol{x}-\boldsymbol{y}_j\right\|^2 \leq \sum_{j=0}^{N-1} P(j|i') \left\|\boldsymbol{x}-\boldsymbol{y}_j\right\|^2,\ \forall i' \neq i \right\} \tag{5.3}$$

and

$$\boldsymbol{y}_j = E\left[\boldsymbol{X}|J=j\right] = \frac{\sum_{i=0}^{M-1} \Pr(I=i)P(j|i)\boldsymbol{c}_i}{\sum_{i=0}^{M-1} \Pr(I=i)P(j|i)}, \tag{5.4}$$

where in (5.4) we defined the *encoder centroids* $\boldsymbol{c}_i = E\left[\boldsymbol{X}|I=i\right]$, and where $P(j|i) = \Pr(J=j|I=i)$ are the transition probabilities of the channel. Generally (see, e.g., [12, 27, 53] and Chapter 2, Section 2.2.1), COVQ design is based on iterating between (5.3) and (5.4) until convergence to a (local) optimum in terms of a stationary point of $D(\mathcal{P},\mathcal{C})$.

We emphasize that the precise model assumed for the discrete channel influences the design only through the transition probabilities $P(j|i)$. Note in particular, that if the channel in Figure 5.1 is used, the decoder codebook has to be larger than the number of encoder indices, effectively resulting in *soft* source (and channel) decoding at the receiver, c.f. [31, 42]. The soft information used by the decoder stems from the additional erasure output symbol of the BSEC. We stress that under the assumptions made and since the decoder codebook is optimal subject to these assumptions, the decoder utilizes the available soft information in an optimal manner.

An alternative interpretation of COVQ's trained for the BSEC, is in terms of multiple description coding. Each received bit can be used to decrease the distortion of the reproduced value, independently of which other bits are received. This is nothing but a special case of multiple description coding, where each bit can be viewed as a description. This topic is further investigated in Chapter 6. Related recent work uses channel optimized scalar quantizers to design multiple description coders for the two channel case [55].

The basic principle of optimizing a COVQ jointly for bit-errors and erasures, that is for $P(j|i)$'s corresponding to the channel in Figure 5.1, is illustrated in Figure 5.2. The figure shows the performance of COVQ's trained for uncorrelated Gaussian data with unit variance. An ordinary

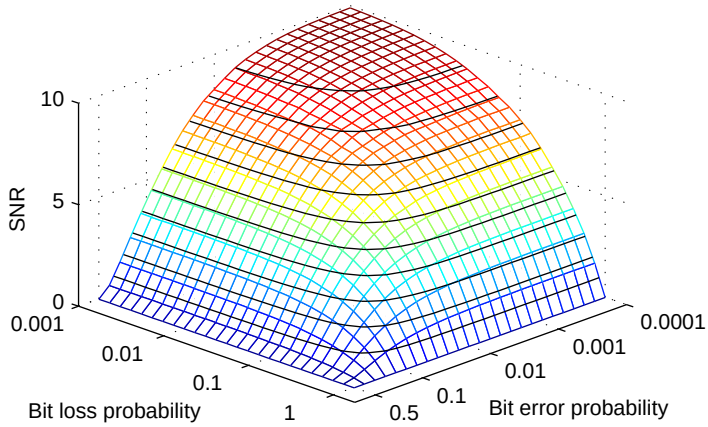


Figure 5.2. Performance of COVQ over different channel parameters.

VQ, trained with the splitting algorithm [29] followed by the binary switching algorithm [54], is used to initialize the COVQ training. In this example, the rate is 2 bits/dimension for 3-dimensional data, and the channel is perfectly matched to the training parameters. The x- and y-axes correspond to different bit error and bit loss probabilities α and β respectively, and the z-axis shows performance in terms of reproduced signal-to-noise ratio $E\|\boldsymbol{X}\|^2/E[d(\boldsymbol{X},\boldsymbol{Y})]$.

5.4 Application to Subband Image Coding

In the present chapter, we will use the basic subband image coder described in Chapter 4, to investigate the performance over channels with bit errors and erasures. As described in Chapter 4, the coder uses a four-level pyramid subband decomposition [3, 35, 39], giving a total of 13 different subbands. Vectors are formed within subbands (no crossband vectors) and each subband is assigned a certain bit rate according to the "equal-slope" method (c.f. [8]). The output indices from the VQ's are put directly in the bit stream, without any additional entropy- or channel coding (except for bit-level interleaving), resulting in a fixed bit rate. Although better compression is possible by using variable-rate entropy coding, we motivate our fixed bit-rate structure by its avoidance of error propagation.

Designing COVQ's for the image coder requires the use of training data. As described in Chapter 4, the approach we use to generate training data is to fit a Gaussian mixture model to the empirical probability density function of vectors from each subband [22]. The model parameters are estimated from a set of images not including the test images. The estimated models can then easily be applied in generating an arbitrary amount of training data. The main reason for using a model of the source distribution instead of training on image data directly, is to avoid over-fitting the COVQ's to a small set of image data.

5.5 Image Results

Figure 5.3. Result from subband coder using regular VQ at 0.5 bits/pixel, designed assuming no channel errors, when subject to 10% bit-losses and 5% bit-errors in the actual channel.

Figure 5.4. Subband coder using COVQ at 0.5 bits/pixel, optimized for 10% bit-losses and 5% bit-errors.

Figure 5.3 shows the result of a subband coder based on ordinary VQ and Figure 5.4 shows the result of a COVQ-based subband coder. In both cases the channel is the binary symmetric erasure channel with 10% erasures and 5% bit errors. For the ordinary VQ case, codewords received with lost bits are reproduced by the unconditional mean of all codevectors that match the partially received codeword. The image results speak for themselves and show a large difference in favor of the COVQ based image coder.

Figures 5.5 and 5.6 show the performance of the same image coders when there are no errors in the transmission. Clearly, the image coder designed under an error free assumption performs better in this case. Of course, the effect is less pronounced than in the case above, but it serves well to illustrate the trade off between designing for good compression and designing for robustness.

Figure 5.5. Result from subband coder using regular VQ at 0.5 bits/pixel.

5.6 Comparison with Forward Error Correction

To compare the performance of the COVQ approach with the performance of a more traditional one, the same image coder structure is used, but with VQ's optimized for the source statistics only. The resulting bit stream from the image coder is then further encoded using BCH codes for protection against errors and erasures. Using *forward error correction* (FEC) increases the total bit rate, so to get a fair comparison, the number of bits allocated to the image coder has to be reduced until the total transmission rate is the same for COVQ and VQ+FEC.

Figure 5.7 shows the performance for three different approaches. One COVQ coder, optimized for BER = 5% and BLR = 10%, and two FEC coders, using BCH codes of length 15 and dimensions 7 and 5 respectively.

Figure 5.6. Subband coder using COVQ at 0.5 bits/pixel, optimized for 10% bit-losses and 5% bit-errors. Received without errors.

The former BCH code can correct 2 errors or 4 erasures and the latter can correct 3 errors or 6 erasures by using ML decoding. In all cases the well known test image Goldhill was encoded to a total bit rate of 0.5 bpp. To demonstrate a channel with both bit errors and erasures, the erasure probability was set to twice the bit error probability for a range of values. The results show that COVQ outperforms FEC for all the simulated channels, even though it is only optimized for *one* specific point on the curve. The flat part of the curves for FEC corresponds to the case when the BCH code can correct *all* errors and erasures. When the code breaks down, the performance drops rapidly. COVQ shows a smoother degradation, observed by others as being typical of COVQ.

Note that it should always be possible to find a COVQ that performs

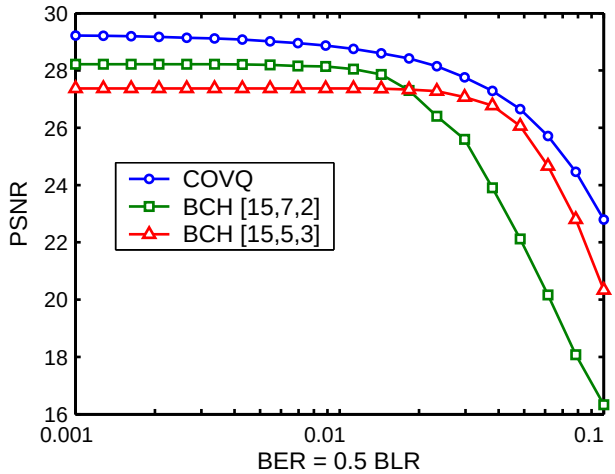


Figure 5.7. Performance of COVQ compared with VQ+FEC

equally well or better than VQ+FEC for any given channel. Such a COVQ can be found trivially, by using VQ+FEC as the initial encoder partitioning in the COVQ training. Since training only can improve performance, one iteration is sufficient to guarantee equally good or better performance.

5.7 Summary

We have presented a simple but important extension of previous work on COVQ for binary memoryless channels with bit-errors to channels with both errors and erasures. Such channels are motivated by a scenario where parts of a network connection are over wireless links. Bit-errors stem from transmission errors and bit-erasures stem from packet losses. An implicit assumption is that no retransmissions are utilized in the network, e.g., due to strong real-time requirements. Hence, packets potentially recognized to contain bit-errors are not declared lost. Instead, packet losses are assumed to be due to other phenomena, such as network overload.

The results show that COVQ is a useful tool for designing joint source-channel codes for the binary symmetric erasure channel. In particular, advantages of joint source-channel coding over separate source and channel coding were demonstrated.

Chapter 6

COVQ-based Multiple Description Coding

In this chapter we present a slightly different approach to multiple description coding than is used in most other papers on MDC. The channel model resulting from the MDC problem description fits perfectly into the framework of channel optimized vector quantization, and an optimal solution to the MDC problem can be identified by inspection.

Even though the traditional formulation of the COVQ design problem obviously holds for general discrete memoryless channels, a major part of the previous works on COVQ have focused on binary symmetric channels. More importantly, only a few previous papers have utilized the connection between the channel optimized and the multiple description quantization problems, including the work by Zhou and Chan [55] and our paper [43]. Zhou and Chan [55] used channel optimized scalar quantizers to make the two-channel multiple description scalar quantization (MDSQ) scheme in [48] robust against symbol errors as well as erasures.

This chapter is arranged as follows: In Section 6.1 we demonstrate that a large class of multiple description design problems can be recast to fit the framework for channel optimized vector quantization (COVQ). Our focus is on problems involving more than two descriptions, since it turns out that the COVQ approach makes these problems relatively simple to implement. In Section 2.3, an index assignment design procedure is presented, that takes advantage of the inherently good index assignment achieved by COVQ training alone. In Section 6.3 we present numerical examples of multiple description

coding with more than two descriptions. The results demonstrate performance gains over previous results on the two description case, while keeping the total transmitted rate constant.

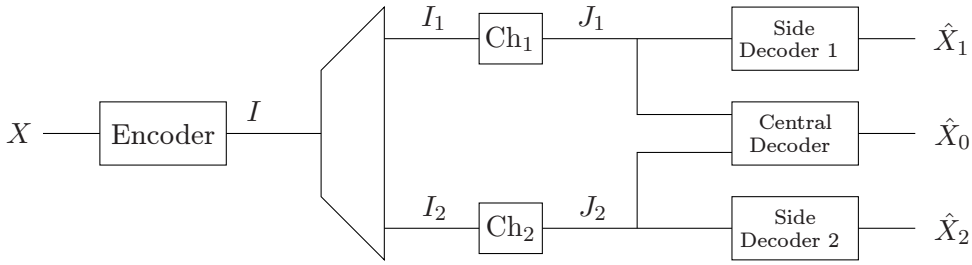


Figure 6.1. Traditional two channel multiple description model

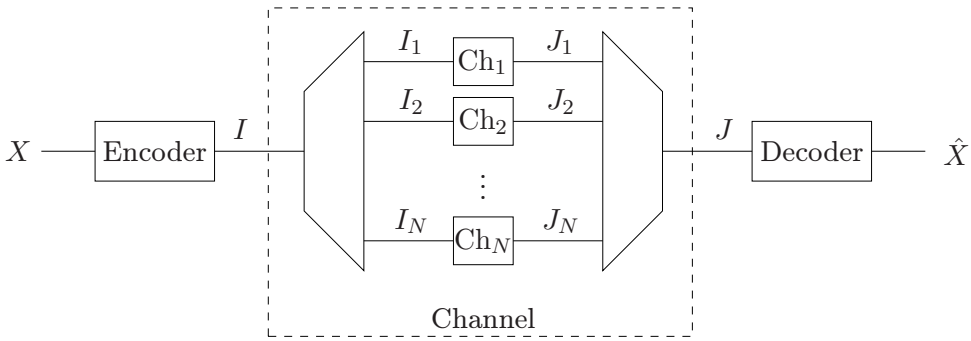


Figure 6.2. Modified multiple description channel model

6.1 Multiple Description Channel Model

Consider the multiple description coding problem as described in Section 2.4. Traditionally, MDC design has been based on having a *set* of decoders, each decoder representing a possible channel breakdown pattern. Figure 6.1 shows this approach for the two-channel case. The design problem is to create an encoder together with a set of decoders, such that the average distortion of each decoder is minimized. This is complicated by the fact that the distortions associated with different decoders are coupled, and the design

problem becomes a multi-criterion optimization problem. It is not clear how to make the trade-off between different decoder distortions.

The standard approach to multi-criterion optimization, is to scalarize the problem as in [48], using Lagrange multipliers to place different weights on different decoder distortions. After scalarization, the problem is readily solved for a local minimum. Typically, several different values of the Lagrange multipliers are tested, until the desired trade-off between the different decoder distortions is achieved.

There are, however, several drawbacks with scalarization. First of all, the properties of the channels do not directly enter into the optimization procedure, but is related in some way to the values of the Lagrange multipliers. Second, and perhaps more important, is that it is difficult to extend the results to systems with more channels than two, due to the rapidly increasing number of Lagrange multipliers needed for the scalarization.

The approach taken in this thesis, is to only consider *one* decoder, and instead extend the symbol alphabet to account for all possible loss patterns. This approach is depicted in Figure 6.2. Note that I and J in this figure belong to different symbol alphabets. If all sub channels are erasure channels, then it is possible to describe the relationship between inputs I and outputs J by the discrete channel transition probabilities $P(J=j|I=i)$. This set of probabilities, together with the source distribution, is all that is needed to write down necessary conditions for the optimal solution using the COVQ framework as given by (2.10) and (2.14).

As an example, consider the simplest possible case: a two-bit codeword split in two one-bit descriptions. If we assume that independent erasures occur with probability p and let $q = (1-p)$, then $P(J=j|I=i)$ is given by:

		J								
		00	01	0x	10	11	1x	x0	x1	xx
	00	q^2	0	pq	0	0	0	pq	0	p^2
	01	0	q^2	pq	0	0	0	0	pq	p^2
I	10	0	0	0	q^2	0	pq	pq	0	p^2
	11	0	0	0	0	q^2	pq	0	pq	p^2

6.2 Index Assignment for MD-COVQ

One issue that is always of great importance in robust quantization, is the problem of *index assignment* as described in Section 6.2. To deal with the complexity of index assignment design, several authors have presented different heuristic approaches to solve the problem suboptimally [12,18,48,52,54]. These methods all have in common that they guarantee that a local optimum can be found at a complexity that is significantly lower than a full search. It is often not very clear how to initialize these methods. The index assignment methods specifically designed for the multiple description problem often require that a number of design parameters must be selected, e.g. the number of redundant encoder symbols to use.

The approach to index assignment preferred by the author of this thesis, takes advantage of a fact that has been observed as being typical for COVQ. Namely that training a COVQ, as described in Section 2.2.4, usually results in a relatively good index assignment. This is true in particular when training for high error probabilities. Another property that turns out to be useful, is that the encoder partitioning after training often has many empty cells, i.e. unused encoder symbols. Having many empty cells significantly reduces the number of possible permutations of the index assignment, since switching place between two empty cells has no impact on the resulting average distortion. This may dramatically shorten the time needed by any additional index assignment algorithm.

A recommended turn-the-crank procedure for the design process is as follows: Start with a codebook generated for an error free channel, i.e. a standard VQ that can be created using e.g. the splitting algorithm [29]. Next, use this codebook to initialize the generalized Lloyd algorithm, and optimize a COVQ for the given multiple description channel. After convergence, the index assignment will be reasonably good and we may choose to stop here. Otherwise we can continue and perform an index assignment algorithm such as the binary switching algorithm (BSA) [54], modified as in [18] to suit the particular MD-channel. If there are many redundant (unused) encoder symbols after the COVQ optimization, then the extra cost for performing the index assignment is relatively low. These steps are summarized in Table 6.1.

Table 6.1. Design steps in codebook generation

1. Select initial codebook
2. Perform COVQ training for the specified channel
3. Optimize the index assignment using the BSA
4. Possibly repeat steps 2. and 3. until no further improvement

6.3 Experimental Results

To demonstrate the performance of the suggested design procedure, a number of simulations on Gaussian data were run. In all simulations, two-dimensional uncorrelated Gaussian data were quantized to 4 bits/dimension, so that the total transmission rate was 8 bits/symbol. These 8 bits were then split into packets of 4, 2 and 1 bit(s), corresponding to 2, 4, and 8 descriptions respectively. The resulting performance is compared with a reference system using conventional VQ, combined with forward error correction (FEC).

Figure 6.4 shows the results of simulations run with quantizers perfectly matched to the channel. In Figure 6.4a, the performance of our suggested design procedure, implemented for the two description case, is compared to the top curve of Figure 4 in [18]. The source distributions, VQ dimensionalities and total bit-rates are identical for both curves. The difference in performance for low probabilities of packet loss, is due to the fact that our suggested procedure (denoted COVQ+BSA) does not put any constraints on the number of redundant encoder symbols, while the design in [18] (MD-BSA) has fixed the number of used encoder symbols to 64 out of 256. Figure 6.4b demonstrates the effect of increasing the number of descriptions, while keeping the total bit-rate constant. The curve corresponding to two descriptions is identical to COVQ+BSA in Figure 6.4a. We see that in this case, a performance gain is possible by splitting the encoder output into more descriptions.

As another reference system, consider a two dimensional VQ with rate R and a Reed-Solomon (RS) code of length N and dimension K. R, N and K are chosen such that $RN/K = 4$ bits/dimension, in order to match the

rate of the COVQ-based system. RS-codes are maximum distance separable (MDS) codes, and the performance can be evaluated as described in [21]. Since the VQ is two dimensional, the number of bits per VQ symbol is $2R$, so the RS code is constructed over GF(2^{2R}). This gives $N = 2^{2R} - 1$, and K is chosen to match the total rate as closely as possible. For instance, $R = 3$ gives 6-bit symbols. The RS code is then constructed over GF(64), $N = 63$ and $K = 47$ gives $RN/K = 4.02$. This comparison is far from fair, since the COVQ system only uses 8-bit codewords, while the RS-code from the example uses $63 \cdot 6 = 378$-bit codewords! Even so, as shown in Figure 6.3, the COVQ system shows better performance for most channel realizations. Note that each FEC-curve has a knee, after which the performance breaks down rapidly. The COVQ-based system shows a much smoother degradation.

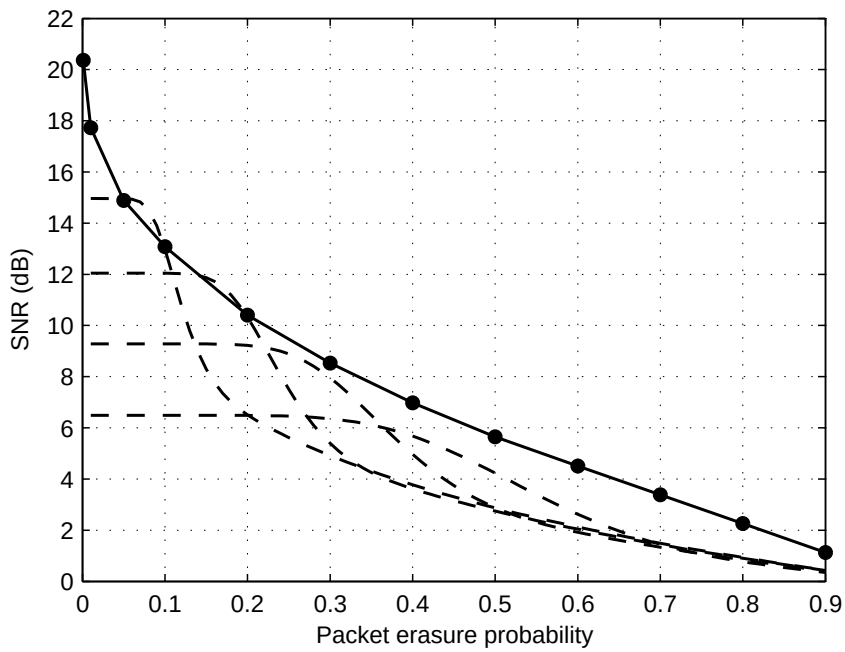


Figure 6.3. Comparison with conventional system

Figure 6.5 shows the sensitivity to channel mismatch in four different scenarios. (a)–(b) show the two descriptions case, and (c)–(d) show the 8 descriptions case. The quantizers in (a) and (c) are trained using COVQ

alone, while those in (b) and (d) have had an extra step of index assignment optimization using the BSA.

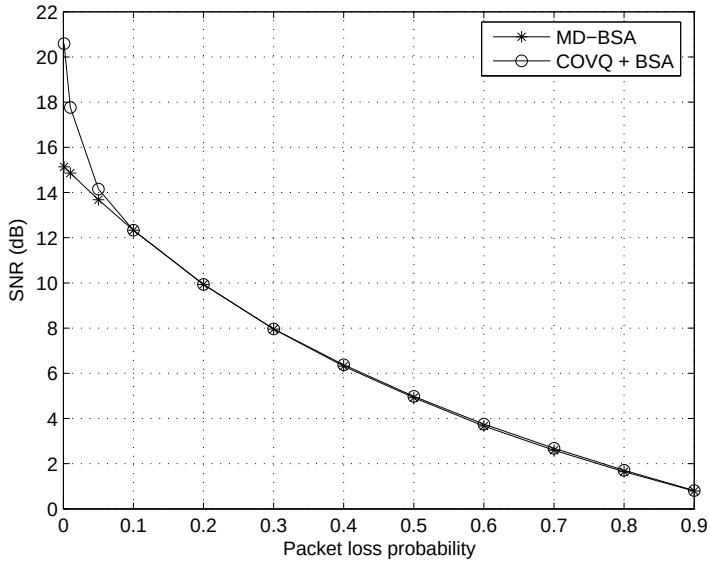
The results in Figure 6.5 show several things worth pointing out. First, it is obvious that the 8 descriptions case is more sensitive to channel mismatch. This implies that if there is any uncertainty in the true channel properties, the gain in performance from increasing the number of descriptions might be lost. Next, we see that using an additional index assignment optimization gives little or no improvement at all if we only consider the envelop of the curves, as we did in Figure 6.4. Index assignment does, however, have a clear positive effect on the robustness of the quantizers. Finally, the quantizer optimized for 10% packet loss in Figure 6.5b has performance that is indistinguishable from the result from [18] shown in Figure 6.4a. It turns out that after COVQ optimization, this particular quantizer makes use of 70 out of 256 encoder symbols, as compared to the 64 used in the example from [18].

6.4 Image Examples

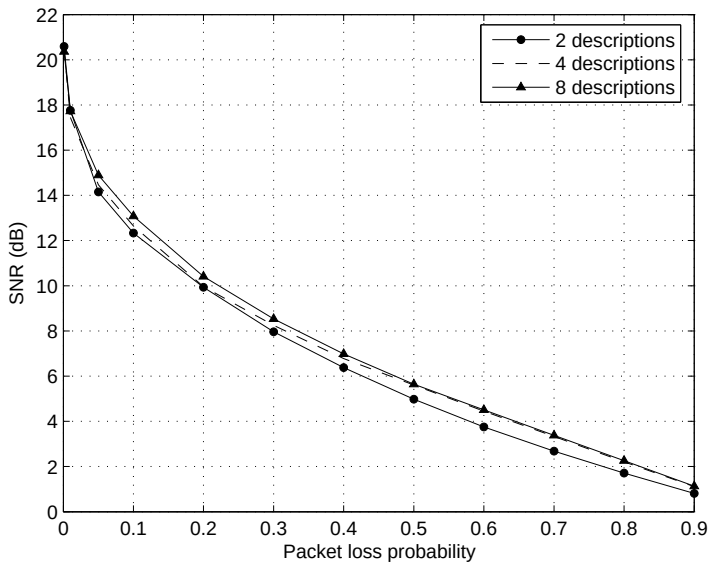
The images in Figures 6.7– 6.8 were encoded using the image coder from Chapter 4. Figure 6.7 was designed using standard VQ without knowledge about the channel. Each bit was transmitted as a separate description. Bit losses were treated by taking the mean value of all possible codevectors matching the partially received index, i.e. if the bit pattern $\{1, \times, 0\}$ was received, the reconstructed value was taken to be the mean value of the codevectors corresponding to $\{1, 1, 0\}$ and $\{1, 0, 0\}$. Figure 6.8 used COVQ trained for the correct channel properties, which in this case was a bit loss probability of 20%.

6.5 Summary

We have presented a simple and straightforward way to design multiple description codes based on channel optimized vector quantization. The suggested approach makes implementation of multiple description codes with more than two descriptions easier than in previous works, and simulation results show that the extensions to more than two descriptions outperform the two descriptions case for channels with high probability of packet loss.

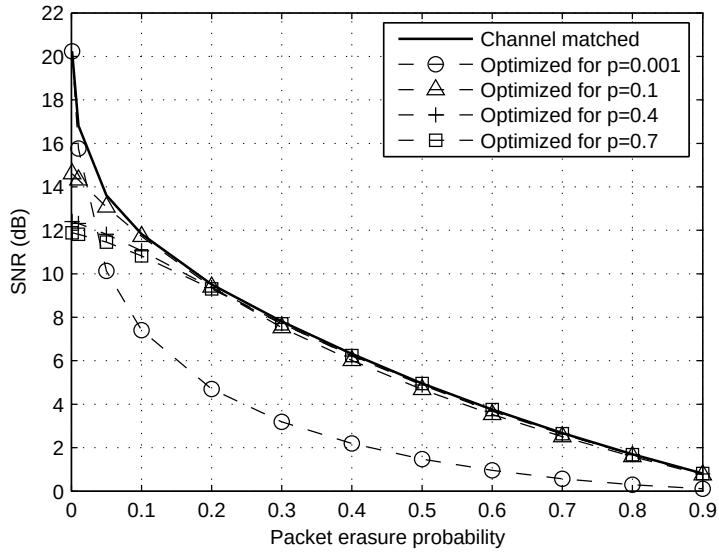


(a)

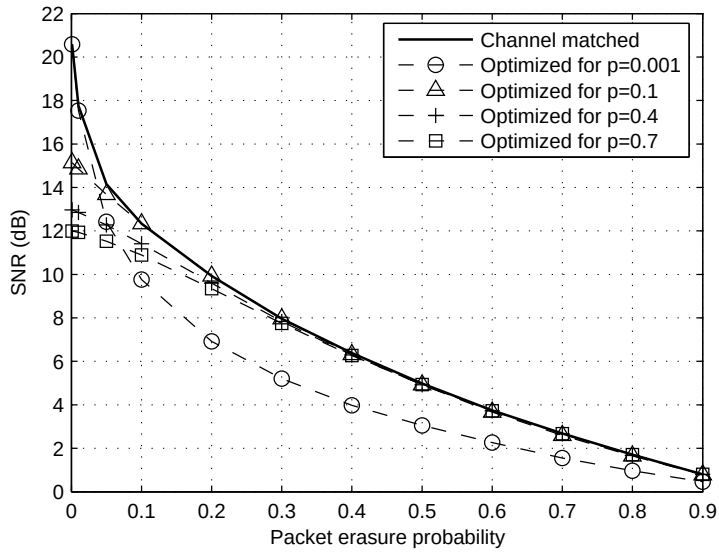


(b)

Figure 6.4. Simulation results for MD-COVQ: (a) 2 description case compared to results in [18]. (b) Performance for different number of descriptions

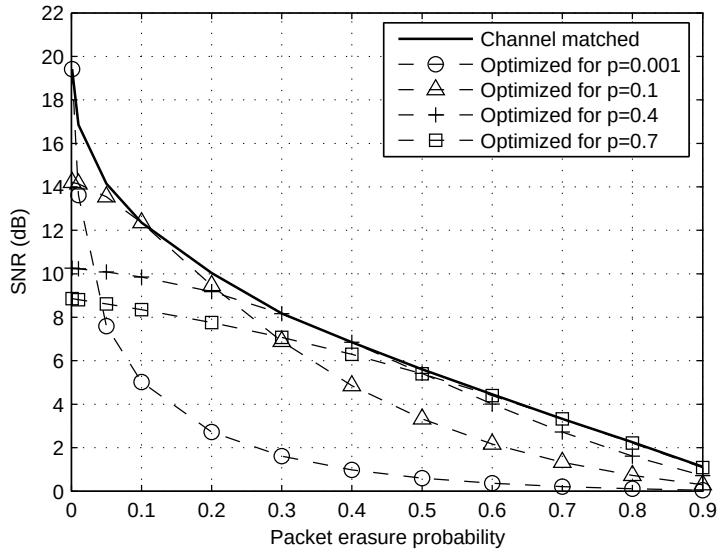


(a)

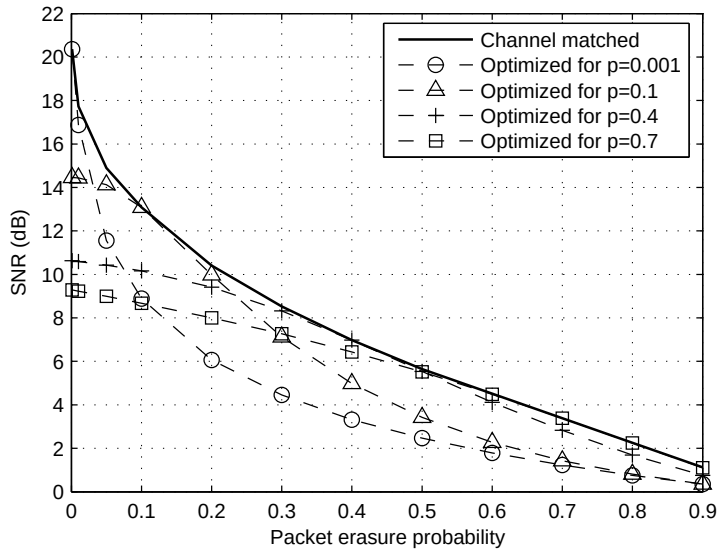


(b)

Figure 6.5. Robustness against channel mismatch. (a) 2 descriptions, COVQ optimized only. (b) 2 descriptions, COVQ with additional index assignment.



(a)



(b)

Figure 6.6. Robustness against channel mismatch, 8 descriptions. (a) 8 descriptions, COVQ optimized only. (b) 8 descriptions, COVQ with additional index assignment.

Figure 6.7. Image encoded using standard VQ. Bit loss probability of 20%.

Figure 6.8. Image encoded using COVQ-based multiple description coding. Bit loss probability of 20%.

Chapter 7

Conclusions

Error robust source coding has been studied in this thesis. Channel optimized vector quantization was used as a tool for two new problems. A new image coder was designed and used for testing and demonstrating the performance of the new schemes. The simulations show that there are potential gains in using joint source–channel coding compared to the traditional approach of separate source and channel coding.

7.1 Future Work

There are a few possible future extensions to this thesis:

- Construct a video coder similar to the image coder in this thesis.
- Study more realistic channel models
  - Internetworking with wireless access
  - Delays, queues, memory
- Study more general network problems in the context of joint source–channel coding
  - Distributed coding and quantization. The Wyner-Ziv and Slepian-Wolf problems.
  - Network coding, "generalized routing" [1].

Bibliography

[1] R. Ahlswede, N. Cai, S. Y. R Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[2] Tomas Andersson and Mikael Skoglund. Design of n-channel multiple description vector quantizers. In *Proceedings Asilomar Conference on Signals, Systems & Computers*, November 2005.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Instrum. Meas.*, 1(2):205–220, apr 1992.

[4] E. Ayanoglu and R. M. Gray. The design of joint source and channel trellis waveform coders. *IEEE Transactions on Information Theory*, 33 (6):855–865, November 1987.

[5] T. Berger. *Rate Distortion Theory — A mathematical basis for data compression.* Prentice-Hall, 1971.

[6] V. Chande and N. Farvardin. Progressive transmission of images over memoryless noisy channels. *IEEE J. Select. Areas Commun.*, 18(6): 850–860, june 2000.

[7] Q. Chen and T. R. Fischer. Image coding using robust quantization for noisy digital transmission. *IEEE Transactions on Image Processing*, 7 (4):496–505, April 1998.

[8] P. C. Cosman, R. M. Gray, and M. Vetterli. Vector quantization of image subbands: A survey. *IEEE Trans. Image Processing*, 5(2):202–225, feb 1996.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[10] P.L. Dragotti, S.D. Servetto, and M. Vetterli. Optimal filter banks for multiple description coding: analysis and synthesis. 48(7):2036 – 2052, July 2002.

[11] J. G. Dunham and R. M. Gray. Joint source and noisy channel trellis encoding. *IEEE Transactions on Information Theory*, 27(4):516–519, July 1981.

[12] N. Farvardin. A study of vector quantization for noisy channels. *IEEE Trans. Inform. Theory*, 36(4):799–809, July 1990.

[13] N. Farvardin and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source–channel coding. *IEEE Transactions on Information Theory*, 33(6):827–838, November 1987.

[14] N. Farvardin and V. Vaishampayan. On the performance and complexity of channel-optimized vector quantizers. *IEEE Transactions on Information Theory*, 37(1):155–159, January 1991.

[15] T. Fine. Properties of an optimum digital system and applications. *IEEE Transactions on Information Theory*, IT-10:443–457, October 1964.

[16] S. Gadkari and K. Rose. Robust vector quantizer design by noisy channel relaxation. *IEEE Transactions on Communications*, 47(8):1113–1116, August 1999.

[17] A. E. Gamal and T. Cover. Achievable rates for multiple descriptions. *IEEE Transactions on Information Theory*, 28(6):851–857, November 1982.

[18] N. Görtz and P. Leelapornchai. Optimization of the index assignments for multiple description vector quantizers. *IEEE Transactions on Communications*, 51(3):336–340, March 2003.

[19] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, 18:74–93, sep 2001.

[20] V. K. Goyal and J. Kovačević. Generalized multiple description coding with correlating transforms. *IT*, 47(6):2199–2224, sep 2001.

[21] V. K. Goyal, J. Kovačević, and M. Vetterli. Quantized frame expansions as source–channel codes for erasure channels. In *Proc. IEEE Data Compression Conference*, pages 326–335, Snowbird, UT, March 1999.

[22] P. Hedelin and J. Skoglund. Vector quantization based on Gaussian mixture models. *IEEE Transactions on Speech and Audio Processing*, 8 (4):385–401, July 2000.

[23] J. Johnston. A filter family designed for use in quadrature mirror filter banks. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 291–294, Denver, CO, April 1980.

[24] P. Knagenhjelm and E. Agrell. The Hadamard transform — A tool for index assignment. *IEEE Transactions on Information Theory*, 42(4): 1139–1151, July 1996.

[25] P. Koulgi, S.L. Regunathan, and K. Rose. Multiple description quantization by deterministic annealing. 49(8):2067 – 2075, August 2003.

[26] J. Kroll and N. Phamdo. Source-channel optimized trellis codes for bitonal imagetransmission over awgn channels. *IEEE Transactions on Image Processing*, 8:899–912, July 1999.

[27] H. Kumazawa, M. Kasahara, and T. Namekawa. A construction of vector quantizers for noisy channels. *Electronics and Engineering in Japan*, 67-B:39–47, January 1984.

[28] A. Kurtenbach and P. Wintz. Quantizing for noisy channels. *IEEE Transactions on Communication Technology*, COM-17:291–302, April 1969.

[29] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Commun.*, 28(1):84–95, January 1980.

[30] D. Mukherjee and S. K. Mitra. A vector set partitioning noisy channel image coder with unequal error protection. *IEEE J. Select. Areas Commun.*, 18(6):829–840, june 2000.

[31] N. Phamdo and F. Alajaji. Soft-decision demodulation design for COVQ over white, colored and ISI Gaussian channels. *IEEE Transactions on Communications*, 48(9):1499–1506, September 2000.

[32] N. Phamdo and N. Farvardin. Optimal detection of discrete Markov sources over discrete memoryless channels — Applications to combined source–channel coding. *IEEE Transactions on Information Theory*, 40 (1):186–193, January 1994.

[33] N. Phamdo, N. Farvardin, and T. Moriya. A unified approach to tree-structured and multistage vector quantization for noisy channels. *IEEE Transactions on Information Theory*, 39(3):835–850, May 1993.

[34] S.S. Pradhan, R. Puri, and K. Ramchandran. n-channel symmetric multiple descriptions-part i:(n, k)source-channel erasure codes. 50(1): 47–61, January 2004.

[35] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, 6(3):243–250, June 1996.

[36] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt. Multiple description wavelet based image coding. *IEEE Trans. Image Processing*, 9(5):813–826, may 2000.

[37] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3 and 4):379–423 and 623–656, July and Oct. 1948.

[38] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. In *IRE National Convention Record*, pages 142–163, 1959.

[39] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing*, 41(12):3445–3462, dec 1993.

[40] M. Skoglund. A soft decoder vector quantizer for a Rayleigh fading channel — Application to image transmission. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2507–2510, Detroit, USA, May 1995.

[41] M. Skoglund. On channel constrained vector quantization and index assignment for discrete memoryless channels. *IEEE Transactions on Information Theory*, 45(7):2615–2622, November 1999.

[42] M. Skoglund and P. Hedelin. Hadamard-based soft decoding for vector quantization over noisy channels. *IEEE Transactions on Information Theory*, 45(2):515–532, March 1999.

[43] Tomas Sköllermo and Mikael Skoglund. A subband image coder for channels with both errors and erasures. In *Proceedings 37th Asilomar Conference on Signals, Systems & Computers*, pages 1553–1557, 2003.

[44] Tomas Sköllermo and Mikael Skoglund. An error-robust image coder for channels with bit errors and erasures. *IEEE Transactions on Communications*, August 2004. in revision.

[45] R. Totty and G. Clark. Reconstruction error in waveform transmission. *IEEE Transactions on Information Theory*, 13(2):336–338, April 1967.

[46] V. Vaishampayan and N. Farvardin. Optimal block cosine transform image coding for noisy channels. *IEEE Trans. Commun.*, 38(3):327–336, mar 1990.

[47] V. Vaishampayan and N. Farvardin. Joint design of block source codes and modulation signal sets. *IEEE Transactions on Information Theory*, 38(4):1230–1248, July 1992.

[48] V. A. Vaishampayan. Design of multiple description scalar quantizers. *IEEE Transactions on Information Theory*, 39(3):821–834, 1993.

[49] R. Venkataramani, G. Kramer, and V.K. Goyal. Multiple description coding with many channels. 49(9):2106–2114, September 2003.

[50] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman. Multiple description coding using pairwise correlating transforms. *IEEE Transactions on Image Processing*, 10(3):351–366, mar 2001.

[51] N. Wernersson, T. Sköllermo, and M. Skoglund. Improved quantization in multiple description coding by correlating transforms. In *Proc. IEEE 2004 6th Workshop Multimedia Signal Processing (MMSP)*, September 2004.

[52] P. Yahampath. On index assignment and the design of multiple description quantizers. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 597–600, Montreal, Quebec, Canada, May 2004.

[53] K. A. Zeger and A. Gersho. Vector quantizer design for memoryless noisy channels. In *Proc. IEEE International Conference on Communications*, pages 1593–1597, Philadelphia, USA, 1988.

[54] K. A. Zeger and A. Gersho. Pseudo-Gray coding. *IEEE Transactions on Communications*, 38(12):2147–2158, December 1990.

[55] Y. Zhou and W.-Y. Chan. Multiple description quantizer design using a channel optimized quantizer approach. In *Proc. of the 38th Annual Conference on Information Sciences and Systems*, Princeton, NJ, March 2004.