Project no: FP6-2002-IST-C 020023-2

Project title: FlexCode

Instrument: STREP

Thematic Priority: Information Society Technologies

D1.3: Video Coder Pilot Study

Due date of deliverable: 2009-03-01

Actual submission date: 2009-07-27 (Updated version)

Start date of project: 2006-07-01 Duration: 36 Months

Organisation name of lead contractor for this deliverable: KTH

Revision: 1.4

Project co-funded by the European Commission within the Sixth Framework Programme 2002-2006		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Abstract

This report describes the technologies developed under the *FlexCode* video coding pilot project. While much of the *FlexCode* methodology is generic in nature, it was designed in the context of scenarios that exist in audio coding. In audio coding the number of variables transmitted in an individual packet is relatively low and a given model family is used to model the signal, whereas in video coding the number of variables is high and the models are selected from various families. This limits the direct applicability of some *FlexCode* source coding techniques to existing video-coding algorithms. Thus, the emphasis of the work in this pilot project was on the application of the *FlexCode* quantization-based multiple-description coding (MDC) methodology to video coding. The *FlexCode* MDC technology is attractive as it asymptotically approach the theoretical performance bounds, contrasting with existing ad-hoc MDC approaches used in video coding. The scalable *FlexCode* quantization-based MDC methods are based on high-rate theory. It was found that this means that energy concentration is vital for good performance. Thus, we studied the use of *FlexCode* MDC in the context of video streaming where a recent spatio-temporal coding architecture that provides a high level of energy concentration. Our experimental results confirm that the *FlexCode* MDC performs well in this context. The new architecture can provide the right level of redundancy for the situation at hand and facilitates bit-plane stripping. The method is particularly useful in the conext of broadcast and multi-cast where the packet-loss rate cannot be specified.

Chapter 1

Overview

1.1 Introduction

The main objective of the *FlexCode* project is to develop *generic* coding technologies that have as distinguishing factor their ability to adapt to the situation at hand instantaneously. The methods avoid application-specific solutions and provide a coding efficiency that is at least similar to current state-of-the-art algorithms. The objective is aimed at a rapidly increasing diversity of services and an increasingly heterogeneous telecommunications network.

While the coding technology developed within *FlexCode* is generic, within the project it is mainly applied to the coding of speech and audio signals. In contrast, this report describes a *pilot project* on the application of *FlexCode source-coding* technology to *video signals*. To see which *FlexCode* technologies are most suitable for video signals, we first investigated the similarities and differences between audio and video coding technologies.

We briefly review the organizational structure and technical approach of the *FlexCode* project to provide a context for this report. The algorithm development work of *FlexCode* is separated into source coding and channel coding. Work package 1 (WP1) deals with source coding and WP2 with channel coding. The work reported here falls in WP1, which has as objective to develop a source coding methodology and its implementation.

By using computable quantizers and probabilistic signal models, the *FlexCode* source-coding methodology allows for instantaneous adaptation of the coding to satisfy rate, quality, and robustness requirements. The model parameters are estimated first, and the quantizers and coding stages can be optimized (computed) based on the statistical models.

The real-time scalable *FlexCode* quantizers are specified by means of equations that are based on probabilistic signal models and knowledge of the channel statistics. The equations have analytic solutions that can be adapted to the quality requirements and network conditions at hand. The source coder can migrate seamlessly from single-description coding to multiple-description coding (MDC) to handle packet loss and can change instantaneously to the optimal level of robustness to the packet loss rate. Bit-plane coding techniques can be used to obtain a layered bit stream that facilitates a reduction in rate after encoding. (In *FlexCode* , the bit-plane techniques cannot be combined with MDC.)

The high-rate theory based source coding approach is complemented with the sensitivity matrix approach used for describing sophisticated perceptual models. The sensitivity matrix approach replaces distortion measures that would make quantization computationally intractable by a local quadratic approximation. This allows the introduction of sophisticated distortion perception-based measures that have hitherto not been used for source coding because of practical problems.

1.2 *FlexCode* Technologies and Audio vs Video Coding

While the *FlexCode* source coding technologies are generic, they rely on the existence of statistical signal models; all coding stages are based on the statistical model. In this section, we discuss whether and where these technologies, which were developed in an audio signal context, are useful for video signals. This requires a discussion of the similarities and differences between speech and audio coding on one side and video coding on the other side. We first discuss the source-coding and then discuss multiple-description coding.

At a high level, audio and video coding paradigms tend to be similar: a pre-processing stage is followed by quantization, which is followed by coding of the quantization indices; at the decoder the matching stages are performed in reverse order. The purpose of the pre-processing is to reduce the dependencies between the variables to be coded. Independence of the variables means that scalar, or low-dimensional, quantization results in only a minor increase in rate compared to the theoretical bounds on coding efficiency. The step of coding the indices can consist of fixed-rate coding (generally trivial), or variable-rate coding (e.g., Huffman coding, arithmetic coding, range coding).

Before we continue, we should extend the simple outline of coding paradigms to include perception. Both in audio coding (e.g., [1, 2]) and video coding (e.g., [3, 4]), it is common to use the pre-processing stage to account for the perception of the signal. Mostly for mathematical and computational convenience, an L^2 distortion measure is commonly used for quantization and the pre-processing can be extended to include a weighting such that the L^2 criterion approximates the perceived distortion. (Under certain conditions it is equivalent to scale the quantizer step sizes.) The signal is weighted by amplifying signal dimensions that are perceptually important and decreasing those that are not perceptually important. In many cases a filter is used for this purpose. Care must be taken that the remaining pre-processing operations do not affect the L^2 distortion measure.

The ability to describe the signal by a *statistical* model forms a vital aspect of the real-time scalable *FlexCode* source coding paradigm. The parameters of the statistical model are estimated for a signal segment. A first innovative aspect of the *FlexCode* paradigm lies in the fact that the pre-processing, quantization, and coding steps are all based on the estimated statistical signal model. The *FlexCode* paradigm assumes that *the propagation of quantization errors is limited* so that a mismatch between encoder and decoder disappears over time if packet loss occurs. A second innovative aspect of the *FlexCode* paradigm is the signal-adaptive transform, which reduces the severity of the approximations made (relative to ideal pre-processing) in the process of creating a practical coder. As *FlexCode* strongly relies on the existence of a statistical model, it is useful to study the notion of modeling in the context of speech/audio coding and video coding.

The standard pre-processing methods aim to reduce dependencies by decorrelating the signal samples either by means of *prediction* and/or by means of *transforms*. The prediction operation is generally adaptive, and the predictor corresponds to a signal model that needs to be known to both encoder and decoder. In speech coding adaptive linear prediction is nearly universal, but it is not so common in audio coding. In video coding the adaptive prediction involves motion compensation. The transform operation of transform coding is generally not adaptive, but relies on the fact the Fourier and related transforms decorrelate stationary signals. This is an approximation. The transform results in a set of variables with nonuniform variance, with the set of variances changing from coding block to coding block. For efficient coding, information about these variances must be known to encoder and decoder. Most common is to simply transmit the maximum value of a set of coefficients; the variance can also be transmitted. This information about the signal variances can be interpreted as the signal model in the context of transform coding.

As mentioned, the pre-processing stage of speech coding usually consists of prediction. The inverse of the prediction-error filter is the autoregressive model filter. The autoregressive signal model describes correlations between the signal samples and, equivalently, the power spectral density of the signal, which is simply the square of the frequency response of the AR model filter. The coefficients of the model are quantized and encoded and the signal samples are decorrelated based on knowledge of the signal model and then quantized and coded, again using knowledge of the signal. Speech coding most commonly uses fixed-rate coding.

Audio coding generally uses the fact that non-adaptive Fourier and cosine transforms can decorrelate stationary signals, asymptotically with increasing block length. The resulting approximation of the power spectral density is divided into frequency bands and the gains of these bands are quantized and coded. The set of gains can be interpreted as a signal model. The signal coefficients are then quantized and coded with knowledge of the gains for the bands. The coding is generally variable-rate.

In the coding of video signals, we can distinguish spatial and temporal pre-processing. Commonly used video-coding paradigms such as H.264, e.g., [5], perform temporal decorrelation with prediction, which is based on determining motion between frames (which location in the previous frame did a pixel come from). The gain of the predictor is generally taken to be one, which means that any mismatch between encoder and decoder does not decay. The motion vectors and the prediction gain can be seen as a signal model. The form of these models varies and this means that we cannot speak of a single model family. This is usually combined with spatial decorrelation based on prediction and fixed transforms on small blocks (8×8 pixels, for example) for each frame, generally in the form of discrete cosine transforms. The transform coefficients are then quantized and the resulting quantization indices are coded. Variable-rate coding is used to encode the coefficients. Note that variable-rate coding is natural when many variables are transmitted simultaneously as the rate averages out. The coding is generally performed simultaneously for all quantization indices of a coding block. Efficient scan patterns for the quantized coefficients are used, followed by context-adaptive binary arithmetic coding. This coding is scalable and effective even when many of the coefficients are quantized to zero. The benefit of context-adaptative coding indicates that dependencies remain after coding. In general these are dependencies that are not removed by a linear operation. No parametric model for these dependencies is provided. In addition, other dependencies are removed with models that are selected from different families in an opportunistic manner. However, the *FlexCode* source coding technology assumes the existence of a fixed model family to remove the dependencies and its scalable quantizers require a rate significantly higher than 1 bit per sample.

A less common class of video coders uses transforms also for the removal of temporal correlations, e.g., [6, 7, 8]. In this case the video frames are represented by a sequence of largely independent images that can be coded with regular image coders. In image coding the wavelet transform is commonly used. The signal is divided into different *scales* and it is possible to identify pixels that describe a particular spatial region at these different scales. It has been found that, for a particular region, the magnitude of the fine-scale coefficient is generally smaller than that of the large-scale coefficients. This suggest the usage of a tree structures (first used by Shapiro [9]) in coding and such trees are indeed commonly used in image coding. The tree structure and the notion of decreasing signal amplitude can be again be interpreted as a signal model.

It is natural to interpret speech and audio coding as the specification of the signal in terms of a model and a description of the signal given the model. Let us look at this in some more detail before we return to video coding. If we make the additional assumption that the signal can be described by a particular family of signal densities (typically Gaussian), then the signal model becomes a statistical model. For the autoregressive model, the signal is interpreted as the result of a convolution of a white Gaussian signal with the autoregressive model filter. For the transform case, depending on the transforms used, the gains correspond to a particular type of convolution in the time domain [10]. Thus, the signal can be seen as the result of this particular type of convolution of the transform of the envelope with a white Gaussian signal. For the autoregressive model, the time samples are distributed according to a multi-variate Gaussian distribution. Since the signal is modeled as stationary, the covariance matrix of the signal is Toeplitz. As the sum of Gaussians is a Gaussian this is also the case for the transform case, but the signal model is cyclo-stationary rather than stationary. For the model description to be meaningful, the convolutions must result in an output that is finite. For the case of the autoregressive model this means that the autoregressive filter must be *stable*. The transform case provides a finite output for any finite input.

In the video coding case, the interpretation of a description in terms of signal model and coding given the signal model is possible but perhaps not as natural as for speech and audio coding. While temporal prediction, including motion compensation, can be seen as a signal model, the unity gain of the filter means that a *FlexCode* approach can not be used as a mismatch between encoder and decoder does not decay. (Naturally error propagation stops whenever an *intra* frame is transmitted; intra-frames were created for this purpose.) While error propagation exists in prediction based video coding, it does not in video coders that use a temporal transform. Although this architecture is not common, it is attractive for broadcast and multicast scenarios.

We can conclude that the generic *FlexCode* technology is not naturally facilitated by the most common video coding architectures. Fine-grained scalability of the encoding accuracy is not a significant advantage. However, the advantages of *FlexCode* MDC may carry over to video coding if its architecture is suitably adapted. In the next section, we discuss how *FlexCode* technology can be used to improve video coding.

1.3 *FlexCode* Multiple-Description Coding for Audio and Video

As the fine-grained scalability of *FlexCode* does not provide a large advantage, we decided to focus on the introduction of the scalable *FlexCode* MDC technology into video coding. MDC fails graciously when the channel capacity is exceeded, whereas the more commonly used error-correcting code based techniques have an all-or-nothing behavior. Thus, MDC has a significant advantage when the exact channel statistics are not known. This is the case if the channel varies or if a range of channels is addressed simultaneously, as is the case in multicast and broadcast scenarios. Note that the fact that averaging is more accurate for the large data quantities of video coding as it is for the smaller data quantities of speech and audio coding does not affect this analysis. In addition, MDC is a form of joint source-channel coding, which means that it generally performs better for shorter coding delays; the separation of source and channel coders is guaranteed to facilitate optimality only for the case of infinite delay.

Effective MDC designs are sensitive to error propagation, and the *FlexCode* system is no exception in this respect. As we discussed in section 1.4, conventional video coding architectures suffer from error propagation. Thus, we studied a number of video architectures on their suitability for MDC before we settled on two specific architectures for more detailed pilot studies.

The first architecture was based on the approach promoted by Jagmohan et al., [11, 12, 13], which uses a distributed source coding [14, 15] to remove the error propagation (encoder-decoder mismatch) problem. For this work, we used the standardized H.264 coder (e.g., [5]) as a video coding platform. The main difference and advantage of our *FlexCode* based approach compared to that of Jagmohan is that we used the scalable *FlexCode* MDC rather than an established MDC (for an overview, see [16]) based on the pioneering techniques of Vaishampayan [17, 18], which are not scalable.

Our distributed-source-coding based MDC (DSC-MDC) work resulted in an M.S. thesis [19] that is available on the *FlexCode* web site. As the coding system is algorithmically complex, the emphasis of the thesis work was to check if the envisioned principles lead to a video coding system that is practically useful. We encountered two issues that limit the practical use of the approach of Jagmohan.

The first problem with DSC-MDC is associated with the high-rate basis for the theory. The implemented system performed as expected for ranges of the rate that are higher than is practically useful in the context of video coding. At lower rates it is difficult to satisfy the high-rate assumptions made in the MDC. In MDC, quantization cells of the so-called side-quantizers each consist of a set of generally disconnected cells of a higher-rate quantizer (the latter is referred to as the *central* quantizer). The large range of values within a single side-quantizer cell means that the distribution of the side-quantizer indices is broader (regions of high probability are now included in more side cells) and, as a result the codewords are longer and the rate is higher. In other words, the signal distribution is relatively narrow compared to the range of the side-quantizer cells and this contradicts the high-rate assumption, leading to inefficient coding.

The sensitivity of DSC-MDC to low-rate effects is inherent to its architecture. In DSC, the distribution of the decoded variables is determined based on side information (consisting of the previous decoded information), thus facilitating a low transmission rate. This operation takes the place of the closed-loop prediction structure in conventional coder architecture. In prediction, the variance of the residual does not vary over time; no explicity energy concentration in time occurs. Similarly, the distribution of the variables in DSC is not time dependent. This lack of energy concentration in time means that at low average rates the above-described low-rate problem extends to many coefficients. While the problem is ameliorated by the spatial energy concentration performed by the discrete cosine transform, a system that concentrates energy in time as well as in space would significantly reduce the low-rate problem. For such a system most coefficients are either high rate, or zero rate.

The second major problem with DSC-MDC is that of rate-control. In distributed source coding the encoder must provide the decoder with sufficient bits to enable decoding. The rate must be sufficient for decoding, but it must also not be too high as we then obtain an inferior rate-distortion performance. In practice, this is often done with "decode-and-request" strategy. However, this approach incurs a long delay.

For more details of our study of the DSC-MDC based we refer to the thesis of Zhe [19], which is available from the *FlexCode* website (www.flexcode.eu).

The second methodology for using *FlexCode* MDC in video coding avoids the problems of the DSC-MDC approach. It is described in more detail in chapter 2. The method concentrates energy both in time and in space. To this purpose the energy-concentrating temporal transform of Flierl [8] is used. The resulting sequence of essentially-independent images is transformed with a wavelet transform to perform spatial energy concentration. The coefficients that form the output of the wavelet transform are quantized with a *FlexCode* MDC quantizer (which is asymptotically optimal). The output of the MDC forms two streams of indices (which correspond to the two descriptions), each of which are losslessly encoded with a conventional image coding algorithm to remove the remaining redundancy. In the particular application we use the SPIHT algorithm for this coding stage. The result is a video coder that can be adopted instantaneously in rate and packet-loss robustness and that has performance that is competitive with the state-of-the-art. The method and its performance is described in more detail in chapter 2

1.4 Report Overview

This report consists of two chapters: this overview chapter, and a chapter describing the practical implementation of *FlexCode* MDC in a video-coding architecture.

Bibliography

[1] J. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Select. Areas Comm.*, vol. 6, no. 2, pp. 314–323, 1988.

[2] B. Edler and G. Schuller, "Audio coding using a psychoacoustic pre- and postfiltering," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Istanbul, 2000, pp. 881–884.

[3] I. S. Hontsch and L. J. Karam, "Locally adaptive perceptual image coding," *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1472–1483, 2000.

[4] K.-T. Lo, X.-D. Zhang, J. Feng, and D.-S. Wang, "Universal perceptual weighted zerotree coding for image and videocompression," *IEE Proc. Vision, Image and Signal Processing*, vol. 147, pp. 261–265, 2000.

[5] I. E. Richardson, *H.264 and MPEG-4 Video Compression*. John Wiley & Sons, 2003.

[6] G. Karlsson and M. Vetterli, "Subband coding of video for packet networks," *Optical Engineering*, vol. 27, no. 7, pp. 574–586, 1988.

[7] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 559–571, 1994.

[8] M. Flierl and B. Girod, "A new bidirectionally motion-compensated orthogonal transform for video coding," in *Proc. IEEE Int. Conf. Aoust. Speech Signal Process.*, vol. I, Honolulu, 2007, pp. 665–668.

[9] J. Shapiro, "Embedded image coding using zerotree of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, pp. 3445–3462, 1993.

[10] P. Korohoda and A. Dabrowski, "Generalized convolution concept based on DCT," in *Proc. EURIPCO*, 20045, pp. 973–976.

[11] A. Jagmohan and N. Ahuja, "Wyner-Ziv encoded predictive multiple descriptions," in *Proc. IEEE Data Compression Conference*, 2003, pp. 213–222.

[12] A. Jagmohan, A. Sehgal, and N. Ahuja, "Predictive multiple description coding using coset codes," in *Proc. IEEE International Conference on Communications Systems*, vol. 2, 2002, pp. 732–737.

[13] ——, “WYZE-PMD based multiple description video codec,” in *Proc. IEEE International Conf. Multimedia and Expo*, vol. 1, 2003, pp. 569–572.

[14] J. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[15] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[16] V. K. Goyal, “Multiple description coding: Compression meets the network,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74–94, 2001.

[17] V. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 39, pp. 821–834, 1993.

[18] V. Vaishampayan and J. Domaszewicz, “Design of entropy-constrained multiple-description scalar quantizers,” *IEEE Transactions on Information Theory*, vol. 40, pp. 245–250, 1994.

[19] D. Zhe, “On distributed video coding with multiple descriptions,” Stockholm, 2008.

Chapter 2

FlexCode Temporal Transform Based Video Coder

G. Zhang, M. Flierl, W.B. Kleijn (KTH)

2.1 Introduction

This chapter describes the *FlexCode* video coding technology that is robust to packet-loss. In contrast to many ad-hoc multiple-description coding (MDC) systems used in video coding, the proposed paradigm can be argued to be near-optimal from a rate-distortion viewpoint. As the proposed coding system is based on high-rate theory, new coder configurations can be created in real-time when the coding environment changes.

Packet loss commonly occurs in packet networks. These losses can be considered as *erasures* in channel transmission. Error-correcting codes and MDC can be used to mitigate the effect of erasures. The usage of MDC can be motivated because it has certain advantages.

Error correcting codes add redundancy to the bit streams. They imply a separation of source and channel coding. Only for long delays is it known that such a separation does not prevent optimal performance. It should be noted that "delay" in this context refers to the number of data transmitted as a block. We can illustrate the "long-delay" result for the case of block codes. A block code is able to correct a certain number of erasures in the block. The redundancy rate associated with the error correcting code corresponds to the theoretical bound (no delay constraint) if it can correct all erasures up to the mean number of erasures in the block. For a probabilistic channel and a finite block size, the number of erasures in a block can exceed the mean number of erasures per block. The law of the large numbers shows that the time-normalized probability of this happening can be made arbitrarily small by increasing the block size.

A problem with the error-correcting code approach is that when it fails, it fails in a catastrophic manner: in this case no information about the transmitted sequence can be retrieved. The impact on distortion of such failures is not considered in the design of error-correcting codes. Thus, it is important to know when the code is likely to fail. Good performance of error correcting codes requires long delay (many data) *and* it requires precise knowledge of the channel. In video coding, where the rate is very high and averaging over many data is possible, the approximation to "long delay" is relatively good. However, precise knowledge of the channel is often not available. An example where this is the case in broadcast and multicast scenarios, where the coded bit-stream is transmitted over a set of channels with varying properties. To avoid catastrophic failure, the rate of the error-correcting code must be set relatively high.

In contrast to error-correcting codes, MDC fails in a soft manner. MDC is a form of combined source-channel coding and good implementations generally perform better at low delays than error-correcting codes. Whereas MDC is optimized using a distortion measure, error-correcting codes are designed without knowledge of the impact of failure to correct the errors. Depending on the number of descriptions of MDC, its quality degrades in a stepwise manner with the number of erasures. The resulting safety margin means that the rate of a good MDC can, at least in principle, be set closer to the theoretical performance bound. Underestimation of the erasure probability may result in a quality that is still acceptable to the user.

In this chapter, we describe an MDC-based video coder that is based on sound theoretical principles and facilitates real-time reconfiguration based on the channel scenario at hand. The architecture facilitates a performance that can approach the theoretical performance bounds for the scalar quantization case. This gives it an advantage over other methods, for which no theoretical analysis exists, e.g., [1, 2].

Temporal transforms are attractive for broadcast and multicast applications because error propagation is not present and because they facilitate the use of quantizer-based MDC. The usage of temporal transforms leads to the subsequent transmission of images that are statistically nearly independent. This means that MDC methods that have been applied to *image processing* are also relevant to our work. Perhaps the most commonly cited image processing MDC system is that of [3]. It uses a decorrelating transform, but is less effective than our system, which is equipped with additional procedures to remove dependencies.

The remainder of this chapter is organized as follows: section 2.2 describes the principles of the approach. Section 2.3 describes the implementation and 2.5 describes the conclusions.

2.2 Coding Principles

To make scalar quantization of signals effective, predictive coding and transform coding are commonly used. The methods remove redundancy by decorrelating the signal samples. In this section, we discuss the choices that led to our video coding architecture.

Transforms, as introduced for the purpose of efficient coding by Huang and Schultheiss in 1963 [4], concentrate energy in relatively few dimensions. Transforms are generally selected to approximate the Karhunen-Loève transform (KLT), which is optimal for certain conditions. While optimality of the KLT is less general than commonly assumed [5], it is optimal for a large range of conditions for variable-rate coding and this suggests that it (and its approximations) forms a good choice for video coding. The KLT diagonalizes the covariance matrix of a signal vector and can be shown to maximize the ratio of the arithmetic and geometric means of the variances of the vector components, which corresponds to a measure of *energy concentration*.

In audio and speech coding the discrete cosine transform (DCT) is commonly used to approximate the KLT; it can be shown that for stationary signals, the DCT approximates the KLT asymptotically with increasing block length. The argument has less of a basis for image and video coding, but the DCT is commonly used. In practice, the wavelet transform often leads to more energy concentration than the DCT. This is not unnatural as images are commonly dominated by edges, and such localized events are associated with energy concentration for wavelets.

Energy concentration is important for the *FlexCode* approach as it facilitates the application of high-rate quantization theory. The assumption that the data density can be approximated as constant within a quantization cell is reasonable for the dimensions that contain most of the concentrated signal energy and its inaccuracy is inconsequential for dimensions with low energy. As will see below, this argument is particularly strong for the MDC case.

While video signals are generally not well approximated by Gaussian densities, analyzing the behavior of transforms and prediction for this density is instructive. The transform coding approach naturally leads to a good approximation of the effect of reverse waterfilling, which is optimal from a rate-distortion viewpoint for Gaussian variables. Reverse waterfiling explains why the failing of high-rate for dimensions with low energy is inconsequential. Reverse waterfilling implies that, at the levels of fidelity required in video and audio coding, it is optimal that many dimensions receive zero rate. Relatively few dimensions have a nonzero rate where high-rate theory is inaccurate. The good behavior of the transform approach contrasts with the predictive coding approach, where the benefit of reverse waterfilling can be obtained by using heuristically optimized postfilters [6, 7]. The postfilters are difficult to optimize, particularly when they are used in combination with perceptual models, preventing adaptation to a new environment or to different requirements.

The case for using transform coding is further strengthed when the usage of quantization-based MDC is envisaged. Quantization-based MDC is attractive as practical implementations can come close to the rate-distortion bounds, contrasting with ad-hoc MDC systems. Two reasons make transform coding desirable for quantization-based MDC. The first is that transform coding does not suffer from the propagation of errors. The propagation of errors in predictive coding means that encoder and the decoder can be mismatched. Since the prediction gain in video coding is typically set to unity, the errors accrue and can lead to severe artifacts. The standard usage of I-frames (intra-coded frames) prevents the long-term accumulation of errors, at the cost of reduced efficiency. The second reason why transform coding is attractive is related to the design-approximations used. Quantization-based MDC generally quantizes the signal with a so-called central quantizer and then splits each index in the sequence, to obtain a number of separate streams (descriptions). Each stream has its own quantization decoder (a so-called side decoder, associated with a side quantizer). In the *FlexCode* approach, each of these side decoders is described with high-rate-theory based approximations. Unfortunately, each of the quantization cells of the side-quantizers generally consists of a set disconnected quantization cells of a so-called central quantizer. This means the side quantizer cells have a large region of support, and that the assumption that the density is constant within the cell is likely to be inaccurate. As a result, the scalar *FlexCode* MDC is not viable if the rate per dimension and per description becomes low (typically a description requires two bits to approximate high-rate theory results). Energy concentration is vital.

As an aside, we note that in video coding so-called *dead-zone* quantizers, which have an increased step size near zero, are common. The high performance of such coders is due to low-rate effects. However, dead-zone quantizers lead to energy concentration (low amplitude signals are quantized to zero) and, as a result, they aid the performance of quantization-based MDC.

In video coding, concentration of signal energy is generally not considered an objective. For each video frame, video coders usually first subtract out a prediction from the previous (quantized) frame, based on motion estimation. (For the H.264 standard this prediction may be replaced by intra-frame prediction.) Note that prediction does *not* concentrate the energy. The prediction operation is followed by a transform operation performed on subblocks (typically anywhere from 4×4 to 16×16 pixels). Thus, the temporal decorrelation is not associated with energy concentration, but the spatial decorrelation is.

In addition to not providing energy concentration, the prediction operator (which generally has a unit gain) also leads to error propagation. Error propagation means MDC is not possible. Jagmohan et al., [11, 12, 13] showed that a distributed source coding [14, 15] approach can be used to avoid error propagation. However, the lack of energy concentration is shared with predictive coding. In addition, distributed source coding suffers from a rate control problem: it is difficult for the encoder to guess the redundancy level required. Thus, we do not consider these methods further.

If we are to maximize energy concentration, we are led to temporal transforms. The usage of temporal transforms is not standardized in video coding. Early systems without [8] and with motion compensation [9] showed the feasibility of the approach but did not lead to broad acceptance. A major reason is likely the delay penalty resulting from the transform. However, in broadcast and multicast scenarios, the coding delay is not a significant disadvantage. In principle, since I-frames are not required, transform coding is more efficient, and it does not suffer from error propagation.

Recently, a new temporal transform approach was introduced that leads to orthogonal transforms that facilitate motion compensation [10]. Using the motion vectors as input, the orthogonal transforms are optimized to maximize energy concentration. The orthogonality of the transforms improves performance over earlier temporal subband systems. In a typical configuration, the temporal transform reorganizes the 16 video-frames into 16 subbands with a low level of redundancy between the bands. (A low redundancy level is loosely associated with the energy concentration.) Each band corresponds to an image (with the same number of pixels as the video signal) that must be transmitted separately. A standard image coder can be used to encode these images. Natural candidates are the wavelet-based zerotree algorithm of Shapiro [11], the SPIHT algorithm [12], or a standardized algorithm such as JPEG2000 [13, 14].

The temporal motion-compensated orthogonal transform (MCOT) of [15, 10] forms a natural basis for robust video coding based on the *FlexCode* quantizer-based MDC. An MCOT-based platform concentrates the signal energy in a small number of dimensions, and does not suffer from error propagation. Combining this platform with the near-optimal performance and flexibility of the *FlexCode* quantizer-based MDC, leads to a robust MDC that is scalable in both robustness and quality. The system is described in the next section.

2.3 Description of the Coding Algorithm

In this section, the architecture of the *FlexCode* coding algorithm is presented. Fig. 2.1 outlines the architecture. The encoding algorithm can be divided into two parts. In the first part, a series of orthogonal transforms are applied to a input video segment: the MCOT and a 2D-wavelet spatial transform. The temporal MCOT aims to remove the inter-frame dependencies. Its output is a set of images for each subsequent block of video frames (for example 8 or 16 frames). The 2D-wavelet transform then reduces the spacial dependencies, separately for each of the images. In the second part, the resulting coefficients are processed to produce one, two, or more encodings (descriptions) of sets of coefficients (image components) that make up the image. Fig. 2.1 illustrates the case with two descriptions. For a given number of encodings, the larger the number of received descriptions, the better the reconstruction quality of an image component. Each of the descriptions is coded with a lossless coder to improve coding efficiency.

The MCOT maintains strict orthogonality with arbitrary motion compensation. In its simplest form it performs a high-band low-band temporal decomposition of a sequence of two images. Let $x_1^{(0)}$ and $x_2^{(0)}$ be two vectors representing consecutive frames of a frame sequence of a video signal.

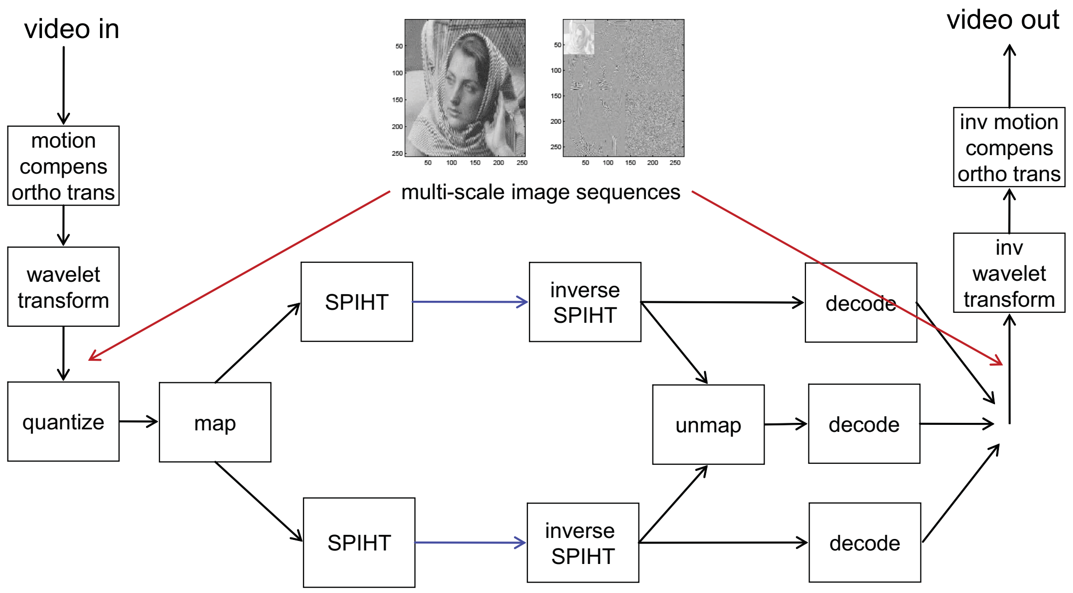


Figure 2.1: *Architecture of the multiple-description video coding.*

A sequence of *incremental* transforms T_κ is then performed such that

$$\begin{pmatrix} x_1^{(\kappa+1)} \\ x_2^{(\kappa+1)} \end{pmatrix} = T_\kappa \begin{pmatrix} x_1^{(\kappa)} \\ x_2^{(\kappa)} \end{pmatrix}.$$

The incremental transform T_κ is orthogonal and differs from the identity matrix in only four elements. It is designed such that $x_1^{(\kappa+1)}$ differs from $x_1^{(\kappa)}$ in one element and $x_2^{(\kappa+1)}$ differs from $x_2^{(\kappa)}$ also in one element. Let these elements be i and j, respectively. Then the transform T_κ is of the form

$$\begin{pmatrix} x_{1,i}^{(\kappa+1)} \\ x_{2,j}^{(\kappa+1)} \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} x_{1,i}^{(\kappa)} \\ x_{2,j}^{(\kappa)} \end{pmatrix}.$$

The pixel $x_{2,j}^{(\kappa)}$ in $x_2^{(\kappa)}$ is selected to be the pixel that is linked to the pixel $x_{1,i}^{(\kappa)}$ in $x_1^{(\kappa)}$ by the motion vector. The incremental transform T_κ tries to remove the energy from the pixel $x_{2,j}^{(\kappa)}$ in $x_2^{(\kappa)}$ by using $x_{1,i}^{(\kappa)}$, under the constraint that T_κ is an orthogonal transform. This energy must then be added to the i'th pixel $x_{1;i}$ in the image $x_1^{(\kappa)}$. All other pixels are not changed. The process is illustrated in Fig. 2.2, which shows the motion vector as d_κ. Subsequent incremental transforms T_κ are selected to cover all pixels of x_2. The full MCOT is $T = \prod_\kappa T_\kappa$. The overall result of the MCOT for a meaningful image sequence is energy concentration into the x_1 component, which can be interpreted as the temporal "low band" component (image) of the motion-compensated video signal. For a static picture, the energy concentration forces all energy into the low-band image.

By performing MCOT on subsequent frames of the same band, a temporal subband decomposition of arbitrary resolution can be obtained (as mentioned before, a typical resolution is 8 or 16 bands). The finer the subband resolution, the longer the delay of the temporal transform.

For each signal block, the output from the MCOT is a set of images, each representing a motion-compensated frequency band of the temporal sequence. These images are then subjected to a conventional wavelet transform. We use the Cohen-Daubechies-Feauveau (CDF) $9/7$ wavelet that is also

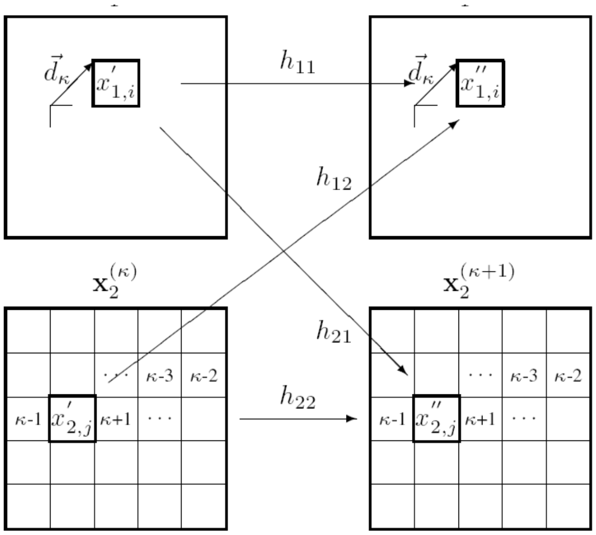

$\vec{d}_\kappa$
$x'_{1,i}$
h_{11}
$\vec{d}_\kappa$
$x''_{1,i}$
h_{12}
$\mathbf{x}_2^{(\kappa)}$
$\mathbf{x}_2^{(\kappa+1)}$
h_{21}
$\cdots$ κ-3 κ-2
κ-1 $x'_{2,j}$ κ+1 $\cdots$
h_{22}
$\cdots$ κ-3 κ-2
κ-1 $x''_{2,j}$ κ+1 $\cdots$

Figure 2.2: *The MCOT.*

used in JPEG2000 [13, 14] for this purpose, but other transforms can be used. For each image this results in a pyramid-like structure of multiple resolutions. As in conventional image coding, the wavelet transform does not remove all dependencies between the coefficients and the next coding stage must exploit the remaining spatial dependencies. In our system this coding stages includes the *FlexCode* MDC.

The *FlexCode* MDC is based on scalar quantization and we perform scalar quantization on the wavelet coefficients. The remaining operations (MDC and removal of remaining dependencies) are performed on the resulting quantization indices. Thus, the scalar quantizer operating on the coefficients forms the *central* quantizer of the MDC system.

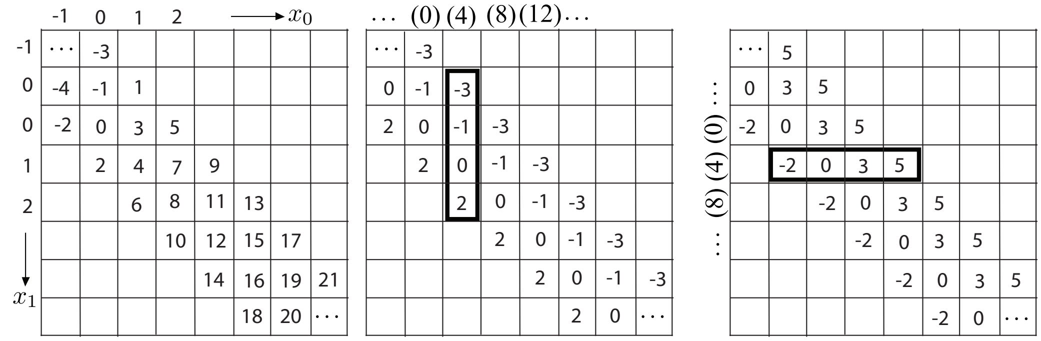
The scalar quantization used in *FlexCode* video coding differs from the systems used in *FlexCode* audio coding. As some variation in rate is generally not a problem for video coding applications, essentially all video coding systems use constrained-entropy coding and the *FlexCode* system is no exception to this rule. High-rate theory indicates that uniform quantizers are optimal for constrained-entropy quantization, independently of the data distribution. However, at low rates dead-zone quantizers, which have an increased step size near zero, perform better (as evaluated with a squared-error criterion). As many dimensions are assigned a low rate, most video coders use uniform quantizers with a dead-zone. As mentioned before, a side benefit for MDC is that dead-zone quantizers concentrate the energy in fewer dimensions.

In constrained-entropy MDC, a higher than strictly needed number of descriptions result in low coding efficiency. Each of the descriptions is individually entropy coded for increased efficiency. However, when a low rate is assigned to a description, as is the case when more descriptions are used, then its quantization cells (which commonly are a set of disconnected cells) tend to have a larger range of support. This in turn makes the probability distribution of the indices more uniform and results in lower coding efficiency. The increased support of the quantization cells leads to increased entropy of the indices of the side quantizer, increasing the rate required for a given quantizer resolution.

For the commonly occurring packet-loss rates below 20 percent generally one or two descriptions suffices. For one description the *FlexCode* system resembles existing MCOT coding systems and so we focus here on the two-description *FlexCode* MDC in the context of a central quantizer with a dead-zone.

The output from the central quantizer is a set of indices that must be split into two sets of indices by the MDC algorithm. The index assignment (IA) matrix plays a central role in this operation. We use a conventional dead-zone quantizer where the deadzone quantization cell is twice the width of the remaining quantization cells. Thus, the central quantizer is a uniform mid-rise quantizer with the two cells closest to the origin combined into one cell. The IA matrix must be designed to account for the dead zone.

Based on the results of [16] (included in deliverable 1.2), we propose a new IA matrix for dead-zone quantizers that has high regularity. The new IA matrix is restricted to have even number of diagonals. As in [16], the nonzero elements of each row are obtained by shifting a fixed pattern by an offset that is a function of the row and column index. Importantly, we use a uniform central quantizer (which is not what is actually used) and for the IA matrix and then note that the indices falling in the deadzone (e.g., -1 and 0) are both mapped into one reconstruction index to obtain the dead-zone central quantizer that is actually used. The regularity of the IA matrix enables the on-line generation of the IA matrix and allows the MDC algorithm to adapt to the environment. The column index of the IA matrix equals the first side quantizer index. However, in contrast to conventional IA matrices, the mapping from the row index to the second side quantizer index is surjective. In effect, the mapping from the row index to the second side quantizer index skips a step around the central cell. Thus, the main difference with [16] is that two rows are associated with the same (side-quantizer) index. Using this setup, the mapping between the indices of the side quantizers and the index of the central quantizer remains *bijective* as is desired for good performance (as central index 0 and 1 are both associated with the deadzone cell) and as is the case for conventional MDC. Note that the two side descriptions are now not perfectly balanced as is usually the case. The row pattern and the column pattern are generally symmetric. Thus, the centroid of each pattern sits in the middle of the pattern. An example of a four-diagonal IA matrix is shown in Fig. 2.3. The column pattern is $[\ -3 \quad -1 \quad 0 \quad 2\ ]$, of which the centroid is -0.5. It is easily seen that the central indices -1 and 0 (together representing the deadzone) both produce column index 0 and row index 0.


-1 0 1 2
x_0
-1
0
0
1
2
x_1
$\cdots$ (0) (4) (8) (12) $\cdots$
$\cdots$ (8) (4) (0) $\cdots$

(a) (b) (c)

Figure 2.3: *The IA matrix for the four-diagonal case. Note that both central index -1 and 0 refer to the deadzone. (a): The original IA matrix. (b): The column pattern after shifting columns. (c): The row pattern after shifting rows.*

After the wavelet transform significant dependencies remain in the coefficients. These dependencies are accounted for in the lossless coding of the side quantization indices. As the number of side quantization indices is large, and as the dependencies are between scales, it is not straightforward to use generic lossless coding approaches, such as arithmetic coding, range coding, or Huffmann coding, to account for these dependencies. It is more natural to use an approach that has been proven to account for dependenceis between wavelet coefficients. To this purpose we treat the indices as coefficient values. It is then possible to use the SPIHT algorithm of Said and Pearlman [12] to perform lossless coding.

It is justified to treat the side-quantizer indices as input to SPIHT since these quantizer indices specify a reconstructed image and since the reconstruction points of the side quantizers correspond to a uniform quantizer or an approximation thereof (as was seen in the discussion of the IA matrix). As the quality of the decoded "side" images increases with increasing redundancy, the dependencies are better retained in the side descriptions when the redundancy level is high. Thus, the dependencies in the wavelet coefficients are most easily exploited in the side descriptions if the MDC is aimed at relatively high packet loss rates. That is, the increased support of the side quantizer cells results in an increased entropy of the vector of side quantization indices. This argument is a straightforward generalization of the earlier argument that the incrased support of the quantization cells leads to increased entropy of the individual indices.

We note that the SPIHT algorithm is inherently a bit-plane coding algorithm. This implies that our MDC facilitates the midstream stripping of bit-planes. Thus, our algorithm allows the real-time optimization of the encoder for the rate and packet-loss rates specified, and it also allows the stripping of bit planes from the transmitted bit stream.

2.4 Configuration and Results

The described *FlexCode* video coding system permits a large range of configurations. The configurations include a trade-off between performance, computational effort, memory requirements, and delay. The first configuration described below is aimed at streaming applications. The main contribution to the computational effort is associated with the motion estimation, which is essential for good performance, but which is not described in this report. However, we can conclude that the settings of the algorithm do not affect overall computational complexity significantly.

We created an MDC system and a reference single-description system (which is optimal for no packet loss). The implementation was set up for the QCIF formatted input (144 × 176 pixels). We used a block size (group) of eight frames for the temporal MCOT. All images resulting from the MCOT were subjected to a four-scale Cohen-Daubechies-Feauveau (CDF) $9/7$ wavelet decomposition. The resulting low-low band region was of size 9×11 pixels. Each pixel in the low-low band is the root of a tree structure of the SPIHT algorithm, resulting in 99 tree structures. These tree structures are then grouped randomly and put into different packets [17]. As a result, each packet can be processed independently at the receiver side, making the coding system more robust. For the experiments, 33 packets were generated for the eight images of a group of frames for the single-description case. Each image contributes three structures to each packet. Thus, each packet contains 24 tree structures, three from every frame. Uniform scalar quantization (with a deadzone) and index assignment mapping are performed to all transformed coefficients to generate two descriptions. By distributing the two descriptions into different packets, 66 packets in total are produced for the two-description case for each eight-frame group.

As input for the experiments we used the video sequence "Foreman". The MDC video coder was first applied to process the video sequence. The average bit rate for each frame was computed and the same average overall rate was imposed on the single-description video coder. The motion information was assumed to be known at the decoder for both the MDC and the single-description video coder.

Both informal viewing and PSNR results confirm that for the same overall rate MDC can perform significantly better than single-description coding when packet losses occur. Informal viewing shows that at 10% and 15% packet-loss rate the two-description based system performs significantly better. Figure 2.4 shows the PSNR performance of the single-description and two-description systems at 10% packet-loss rate. The improvement when using MDC is similar at 15% packet-loss rate. As expected, at zero packet-loss rate the single-description system performs visibly better. Somewhat surprisingly, in terms of PSNR the advantage of the single-description system over the MDC system at zero packet loss is only about 1.5 dB.

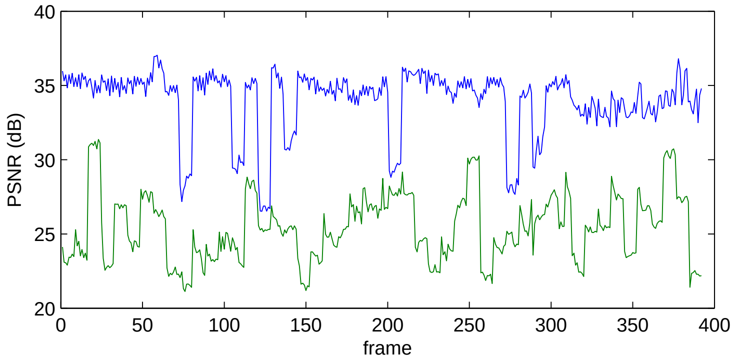


Figure 2.4: *The PSNR for the Foreman sequence for two systems. Excluding motion information, the bit rate is 0.68 bits per pixel and the packet loss rate is 10% for both systems. The better performing signal is the MDC system, and the other system is the single description system.*

2.5 Conclusions

We have shown how the scalable, quantizer-based *FlexCode* MDC [16] (see also deliverable 1.2) can be integrated in a temporal-transform based video coding system aimed at streaming applications. The algorithm can be optimized instantaneously to match the rate and packet-loss rate specified, and bit-planes can be dropped from the streams anywhere during the transmission process. The advantage of the *FlexCode* MDC method over other MDC methods used in video coding is that the *FlexCode* method, as it is quantization-based, is asymptotically optimal. The relatively high rates per dimension required for the good performance of such a system are naturally provided by a coding architecture that concentrates the energy (rendering few high-rate and many zero-rate dimensions). This means that the *FlexCode* MDC is a natural companion for the motion compensated orthogonal transform introduced in [15, 10]. The experimental results confirm that the system provides good performance. The arrangement is particularly attractive for situations where the channel descriptions are inherently fuzzy, such as broadcast and multicast. For these applications, it likely is beneficial to extend the system to include asymmetric MDC systems.

Bibliography

[1] M. Biswas, M. Frater, and J. Arnold, “Multiple description wavelet video coding employing a new tree structure,” *IEEE Circuits Systems for Video Techn.*, vol. 18, no. 10, pp. 1361–1368, 2008.

[2] D. Wang, N. Canagarajah, D. Redmill, and D. Bull, “Multiple description coding based on zero padding,” in *Proc. ISCAS*, 2004, pp. 205–208.

[3] S. D. Servetto, K. Ramchandran, V. Vaishampayan, and K. Nahrstedt, “Multiple description wavelet based image coding,” *IEEE Trans. Image Process.*, vol. 9, no. 50, pp. 813–826, 2008.

[4] J. Huang and P. Schultheiss, “Block quantization of correlated Gaussian random variables,” *IEEE Trans. Comm. Syst.*, vol. 11, pp. 289–296, 1963.

[5] M. Effros, H. Feng, and K. Zeger, “Suboptimality of the Karhunen-Loève transform for transform coding,” *IEEE Trans. on Inform. Theory*, vol. 50, pp. 1605–1619, 2004.

[6] S. V. Andersen and W. B. Kleijn, “Reverse water-filling in predictive encoding of speech,” in *IEEE Speech Coding Workshop*, Porvoo, 1999, pp. 105–107.

[7] M. Y. Kim and W. B. Kleijn, “KLT-based adaptive classified vector quantization of the speech signal,” *IEEE Trans. Speech Audio Process.*, vol. 12, no. 3, pp. 277–289, 2004.

[8] G. Karlsson and M. Vetterli, “Subband coding of video for packet networks,” *Optical Engineering*, vol. 27, no. 7, pp. 574–586, 1988.

[9] J.-R. Ohm, “Three-dimensional subband coding with motion compensation,” *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 559–571, 1994.

[10] M. Flierl and B. Girod, “A new bidirectionally motion-compensated orthogonal transform for video coding,” in *Proc. IEEE Int. Conf. Aoust. Speech Signal Process.*, vol. I, Honolulu, 2007, pp. 665–668.

[11] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[12] A. Said and W. Pearlman, “A new, fast, and efficient image codec based on set partioning in hierarchical trees,” *IEEE Trans. Circuits and Syst. Video Techn.*, vol. 6, no. 3, pp. 243–250, 1996.

[13] C. Christopolous, A. Skodras, and T. Ebrahimi, “The JPEG2000 still image coding system: an overview,” *IEEE Trans. Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2000.

[14] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, “An overview of JPEG-2000,” in *Proc. Data Compression Conference*, 2000, pp. 523–544.

[15] M. Flierl and B. Girod, “A motion-compensated orthogonal transform with energy concentration constraint,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Victoria, BC, 2006.

[16] G. Zhang, J. Klejsza, and W. B. Kleijn, “Optimal index assignment for multiple description scalar quantization,” 2008, submitted for publication.

[17] J. K. Rogers and P. C. Cosman, “Wavelet zerotree image compression with packetization,” *IEEE Signal Processing Letters*, vol. 5, pp. 105–107, 1998.