Multiple Description Coding for Video Delivery

YAO WANG, FELLOW, IEEE, AMY R. REIBMAN, FELLOW, IEEE, AND SHUNAN LIN, MEMBER, IEEE

Invited Paper

Multiple description coding (MDC) is an effective means to combat bursty packet losses in the Internet and wireless networks. MDC is especially promising for video applications where retransmission is unacceptable or infeasible. When combined with multiple path transport (MPT), MDC enables traffic dispersion and hence reduces network congestion. This paper describes principles in designing MD video coders employing temporal prediction and presents several predictor structures that differ in their tradeoffs between mismatch-induced distortion and coding efficiency. The paper also discusses example video communication systems integrating MDC and MPT.

Keywords*—*Multipath transport, multiple description coding (MDC), video communications.

I. Introduction

Multiple description coding (MDC) has emerged as a promising approach to enhance the error resilience of a video delivery system. In the most common implementation, a multiple description (MD) coder generates two equal rate and equal importance descriptions so that each description alone provides low but acceptable quality and both descriptions together lead to higher quality. The two descriptions are individually packetized and sent through either the same or separate physical channels. As long as the two descriptions are not simultaneously (in terms of the spatial location and time in the underlying video sequence) affected by packet losses, an acceptable quality can be maintained. In the more general case, more than two descriptions can be generated, which may or may not have identical rates.

A primary reason for the increasing popularity of MDC is that it can provide adequate quality without requiring retransmission of any lost packets (unless the loss rate is very high). This has two important implications. First, it makes MDC particularly appealing for real-time interactive applications such as video phone and conferencing, for which retransmission is often not acceptable because it incurs overlong delay. Second, it simplifies the network design: no feedback or retransmission is necessary and all the packets can be treated equally. This is in contrast to layered coding (LC), which generates a base layer and one or more enhancement layers. One major obstacle for the adoption of LC in practical networks is that to guarantee a basic level of quality, the base layer must be delivered almost error free. This requires differential treatment of base-layer and enhancement-layer packets by the network and retransmission of lost base-layer packets. Depending on the acceptable delay of the underlying application and the network setup, these options are not always feasible.

In addition to error resilience, combining MDC with multiple path transport (MPT) enables traffic dispersion and load balancing in the network, which can effectively relieve congestion at hotspots and increase overall network utilization. Although one can split the bitstream from any coder onto multiple paths, the fact that substreams from MDC can be treated equally and independently makes the task of allocating and scheduling source packets onto transport paths much easier than for a conventional single description (SD) coder or a layered coder. Combining MDC with MPT has become a "hot" new research direction in the past few years because this architecture offers both error resilience and load balancing and is applicable in both wired and wireless networks.

These benefits of MDC come at a price, however; for an MD coder to meet the same quality criterion as a conventional SD coder in the absence of any transmission errors, the MD coder uses more bits. This *excess rate* or *redundancy* is inserted intentionally to make the bitstream more resilient to transmission errors. The primary objective in designing an MD coder is to minimize the redundancy (or the total rate) while meeting an end-to-end distortion requirement that takes into account transmission loss.

The idea of MDC was conceived and studied in the early 1980s by information theorists [1]–[3]. Some MD coders

Manuscript received January 11, 2004; revised May 20, 2004. This work was supported in part by the National Science Foundation under Grant ANI 0081375.

Y. Wang is with the Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY 11201 USA (e-mail: yao@duke.poly.edu).

A. R. Reibman is with AT&T Labs.—Research, Florham Park, NJ 07932 USA (e-mail: amy@research.att.com).

S. Lin is with Harmonic Inc., White Plains, NY 10601 USA (e-mail: shunan.lin@harmonicinc.com).

Digital Object Identifier 10.1109/JPROC.2004.839618

were later proposed, including [4] and [5]. In the second half of the 1990s, MDC became popular as an effective means to combat transmission errors in the best-effort Internet and wireless networks. Numerous MD coders have since been proposed for coding multimedia (speech, image, and video). For a comprehensive review of the development history of MDC, the rate-distortion (R-D) bound for MDC, and various MD algorithms developed (primarily for images), the reader is referred to Goyal's overview paper [6].

MDC is particularly promising for video because of the very stringent delay requirement in many video applications. However, designers of MD video coders face some unique challenges. As is well known, motion-compensated prediction can effectively exploit the temporal correlation between video frames. As a result, it is a fundamental component in all video coding standards. Therefore, many MD video coders also incorporate motion-compensated prediction. However, whenever an encoder uses a signal for prediction that is unavailable to the decoder (because of transmission loss), a condition called "mismatch" occurs. MD coding is designed around the principle that information may be lost during transmission; hence, mismatch is a fundamental design concern in MD video coders. Efficient predictors are more likely to introduce mismatch, so a primary technical challenge is to balance prediction efficiency with mismatch control.

In Section II, we describe the principles of designing MD video coders. In Section III, we review some representative implementations of MD video coders based on the strategies they use for MD coding, prediction, and mismatch control. Section IV presents the general system architecture that integrates MDC and MPT and reviews three representative systems. Section V summarizes the main contributions of this paper and suggests several future research directions.

II. Principles of Predictive MD Coding

In this section, our focus is on describing two fundamental design issues that must be tackled when creating useful MD video coders: mismatch control and redundancy allocation. We restrict our discussion to the two-description case.

We discuss general design strategies for an MD coder sent on an ideal MD network in Section II-A. In predictive coders in general, the encoder is designed to ensure that the predictions formed at both the encoder and decoder always match. In a predictive MD (P-MD) encoder, this may incur redundancy; hence, it may not always be desirable to eliminate mismatch. Section II-B examines P-MD coders and considers alternatives for handling mismatch in an ideal MD network. Section II-C highlights the difference between the ideal MD network and practical packet networks and presents alternatives for mismatch control in such networks.

In general, overall performance of an MD coder depends on the channel characteristics and the source statistics. Since both these vary with time for many practical video applications, it is important that MD coders be designed with the ability to be flexible and dynamically adapt the amount and

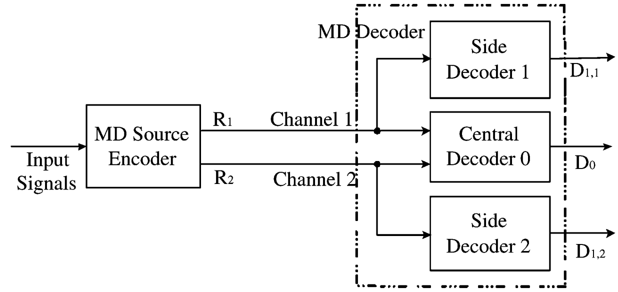


Fig. 1. MD source coding for an ideal MD network.

types of redundancy. Section II-D discusses redundancy allocation in a P-MD coder.

A. Characterization of a General MD Coder

An ideal MD network consists of two channels. Either channel may fail with probability $p_i, i = 1, 2$. If a channel fails, it does not work throughout the duration of the transmission. The decoder knows when a channel has failed, while the encoder cannot know. A typical assumption is that both channels do not simultaneously fail.

Fig. 1 shows the basic framework for MD source coding using an ideal MD network. The encoder creates two descriptions which are sent separately across two channels. The bit rates used to send each description, in bits per source sample, are R_1 and R_2, and the total rate is $R = R_1 + R_2$. Three situations are possible: both descriptions are received by the MD decoder or either one of the two descriptions is missing.

It is conceptually useful to describe the basic MD decoder as consisting of three individual decoders, each corresponding to the three situations. This forces the encoder to consider explicitly that the decoder may be in one of three states, even though the encoder cannot know which of the three states the decoder is in. The central decoder receives both descriptions and produces a high-quality reconstruction with central distortion D_0, while the two side decoders each receive only one of the two descriptions and produce lower, but still acceptable, quality reconstructions with side distortions $D_{1,1}$ and $D_{1,2}$. In many applications, a balanced design is useful, where $R_1 = R_2$ and $D_{1,1} = D_{1,2} = D_1$ and $p_1 = p_2 = p$.

An SD coder minimizes D_0 for a fixed total rate R, and its performance is measured by its operational rate-distortion function $R(D_0)$. An MD coder has conflicting requirements to simultaneously minimize both D_0 and D_1. At one extreme, an MD coder that simply alternates the R bits of an SD bitstream into each description will achieve the minimum central distortion but have unacceptably high side distortion. At the other extreme, an MD coder that simply duplicates the SD bitstream with rate R into each description will use $2R$ bits to achieve minimal side distortion but have a larger central distortion compared to that of an SD coder using $2R$ bits.

One way to measure the efficiency of an MD coder is by using the redundancy-rate distortion (RRD) curve [7]. Define the distortion of the best single description (SD) coder to be D_0 when R^* bits are used. Then, define redundancy to be $\rho = R - R^*$, where R is the rate when the MD coder has central distortion D_0. Intuitively, ρ is the bit-rate sacrificed compared to the SD coder for the purpose of reducing D_1. The RRD function $\rho(D_1; D_0)$ is the additional rate which is required to achieve a desired side distortion D_1 at central distortion D_0. The performance of an MD coder is characterized by three variables: R, D_0, and D_1.

Let $\mathcal{M}$ be the set of parameters that define the MD coder that is based on an SD coder with parameter set $\mathcal{Z}$.[1] When optimizing an MD encoder for transport over an ideal MD network, a common approach is to minimize the average distortion based on the (assumed known) channel failure probabilities subject to a rate constraint. This leads to

$$\min_{\mathcal{Z},\mathcal{M}}((1-p)^2 D_0 + 2p(1-p)D_1 + \lambda R) \tag{1}$$

$$\text{subject to } R \le R_{\max}. \tag{2}$$

We note that the central distortion depends on $\mathcal{Z}$ only, and the side distortion depends on both $\mathcal{Z}$ and $\mathcal{M}$. The total rate R can similarly be expressed as $R = R^*(\mathcal{Z}) + \rho(\mathcal{M}, \mathcal{Z})$. Then, (1) can be written as

$$\min_{\mathcal{Z}}\{(1-p)^2 D_0(\mathcal{Z}) + \lambda R^*(\mathcal{Z}) + \min_{\mathcal{M}}[2p(1-p)D_1(\mathcal{Z},\mathcal{M}) + \lambda\rho(\mathcal{M},\mathcal{Z})]\}. \tag{3}$$

The above optimization problem can be considered a problem of allocating a total rate $R_{\max}$ between R^* and ρ to minimize the average distortion. If the central distortion is determined by the application, $\mathcal{Z}$ and R^* are fixed. Then, only the inner optimization, which corresponds to finding the best point on the RRD function, is necessary.

B. Predictive Coding in an MD Environment

Because predictive coding is nearly universal in today's video coders, we consider generic P-MD coders. The generic P-MD encoder and decoder we present here is meant to be illustrative rather than exhaustive or comprehensive. Details on how specific implementations are built from or may vary from these simple structures are described in Section III. We focus on the predictive structure, recognizing that intraframes are a special case where the predictor is zero.

Consider a predictive coder in a single-description environment. The encoder typically tracks the state S it expects to be present at the decoder and bases its predictor P upon that state. In this way, encoder and decoder are able to maintain identical state, provided the decoder receives all information.

Fig. 2 shows a simplified generic MD predictive decoder. The central and side decoders inside the dotted boxes are each typical predictive decoders. The two descriptions contain prediction error information. As before, the central decoder is used when both descriptions are received, and the side decoders are used when only one description is received. The prediction error signals are decoded by the MD prediction-error (MD-PE) central and side decoders. Each of the predictive decoders has its own predictor function P_i based on the state, $S_i, (i = 0, 1, 2)$, available to that decoder. The

[1]For example, for both the MDSQ coder [5] and the MDTC coder [7], the set $\mathcal{Z}$ is the quantizer parameters. For MDSQ, $\mathcal{M}$ corresponds to the MD index assignment, and for MDTC, $\mathcal{M}$ is the MDTC transform matrix.

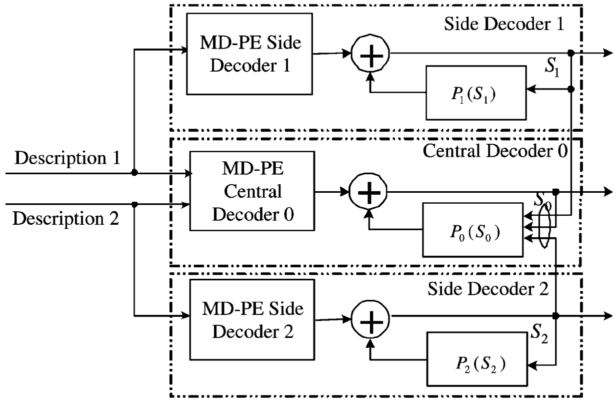


Fig. 2. Predictive MD decoder.

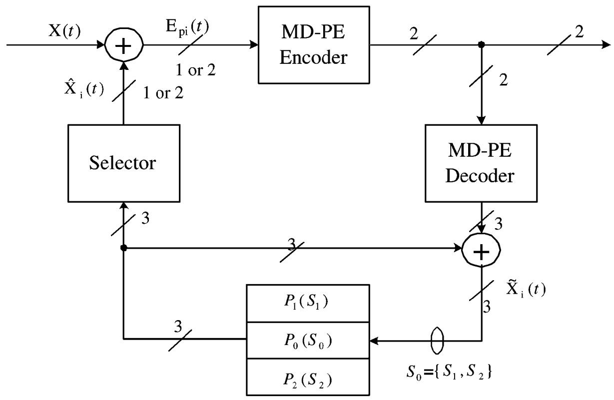


Fig. 3. Predictive MD encoder.

state of the central decoder, S_0, includes the state of the individual side decoders. The central predictor function P_0 may not use all components of S_0. The outputs of the side decoders and the central decoder may be processed by an additional block (not shown) prior to display to improve quality.

Depending on which descriptions are received, the decoder has three possible states, S_0, S_1, and S_2. However, the encoder can never know which of these states is present. If the encoder uses a predictor that depends on state not available at the decoder, the encoder and decoder states will be mismatched. This potential mismatch and the subsequent error propagation present a fundamental design concern for P-MD video coders.

In general, a P-MD encoder must produce two descriptions that will be useful to the decoder, regardless of which state it is in. A variety of strategies have been proposed, with different tradeoffs between redundancy and side distortion. Fig. 3 shows a simplified generic P-MD encoder. The structure is similar to a basic predictive coder, with an MD-PE encoder in place of the usual quantizer and an MD-PE decoder in place of the usual inverse quantizer. Each branch (arrow) may contain one, two, or three signals, as indicated. In some implementations, some branches may be omitted or additional branches added, as described in more detail. The generic encoder of Fig. 3 tracks all three states in the decoder of Fig. 2. To keep the block diagram simple, we specifically omitted any optional block for coding the mismatch signal.

Table 1
Summary of Predictor Classes

Predictor class	Definition
A	Predictor(s) that introduce no mismatch
B	Single-description predictor; (no prediction inefficiency, but with mismatch)
C	Predictor that controls trade-off between prediction efficiency and mismatch

The generic encoder operates as follows. Depending on the implementation, the encoder typically has either one or two basic predictors. Let $\hat{X}_i(t) = P_i(S_i)$ be the ith prediction of the original signal $X(t)$. If the encoder has only one predictor, it is $\hat{X}_0(t) = P_0(S_0)$, which may not necessarily be the same predictor as a single-description coder. If the encoder has two predictors, they are $\hat{X}_1(t) = P_1(S_1)$ and $\hat{X}_2(t) = P_2(S_2)$. The ith prediction error signal is $E_{p,i}(t) = X(t) - \hat{X}_i(t)$.

The MD-PE encoder can be implemented using a wide variety of techniques, to be described in Section III-A. In general, it takes as input the prediction error signal(s) and adds redundancy to create two correlated signals, one for each description. The MD-PE decoder has three outputs, one from the central MD-PE decoder and two from the side MD-PE decoders. Each output from the MD-PE decoder is added to its respective prediction signal to form the reconstruction $\tilde{X}_i(t)$.

Mismatch will occur in an ideal MD network whenever the states used for prediction at the encoder and decoder do not match. The error due to mismatch in one frame will propagate into subsequent frames, and the resulting increasing error as a function of time may become disturbing. Therefore, P-MD coders should be designed to be fully aware of any mismatch they may introduce.

The reconstruction errors for the three decoding states are $E_{r,i}(t) = X(t) - \tilde{X}_i(t), i = 0, 1, 2$. The mean square of $E_{r,0}(t)$ defines the central distortion D_0, while that of $E_{r,i}(t), i = 1, 2$ defines the side distortion D_1. To emphasize the presence of the mismatch error in the side distortion for a P-MD coder, we assume the mismatch error is uncorrelated with the rest of the error[2] and express the total side distortion as

$$D_1 = D_{1,\text{perr}} + D_{1,\text{mis}}. \tag{4}$$

$D_{1,\text{perr}}$ is the side distortion in representing the prediction error by the MD-PE encoder, and $D_{1,\text{mis}}$ is the side distortion due to mismatch.

The predictor block shown in Fig. 3 outputs three prediction signals to mimic the prediction signals created in three different decoding states. In some implementations, not all prediction signals are generated. We categorize the possible predictors for an P-MD coder into three classes, depending on the resulting tradeoff between overall redundancy and the side distortion. These categories are summarized in Table 1.

[2]This assumption is not accurate in coders with explicit mismatch coding.

Class A) Predictors in Class A have no mismatch. This requires the encoder to create a prediction error signal using the same prediction signal(s) as the decoder, regardless of which state the decoder is in.

There are two principle ways for an encoder to achieve this. The first is to use two individual predictors, $\hat{X}_i(t) = P_i(S_i), i = 1, 2$, that predict based only on information sent in one of the two descriptions. Such an encoder can be thought of as a "two-state" encoder [8]. Some examples of coders built using this approach are [9], [8], [10], and [11]. These are explained in more detail in Section III-B.

The second way is for the encoder to use a single predictor that uses information common to both descriptions. MD-SNR [12] is one example, as is the coder in [13]. These are explained in more detail in Section III-C.

Class B) Predictors in Class B use the same predictor that would be used by a single-description (SD) predictive encoder. This predictor minimizes the prediction error and hence introduces no additional redundancy during the prediction process. In general, the decoder cannot form this prediction unless both descriptions have been received; hence, mismatch will occur. Many MD video coders use this predictor strategy; a few examples are [14], [15], and [11]. These coders and others are explained in more detail in Sections III-D and III-E.

Class C) A predictor in Class C has parameters that can be selected appropriately to trade the loss in prediction efficiency (compared to an SD predictor) and the amount of mismatch. Two examples of encoders using this class of predictor are [16] and [17]. These are explained in more detail in Section III-F.

Because predictors in Classes B and C introduce mismatch, an encoder using one of these predictors can optionally send a compressed version of the mismatch. The side decoder, by combining the central prediction error and the mismatch signal, can completely or partially cancel the prediction mismatch when reconstructing each frame. Another approach to control mismatch is to periodically send I-frames, which clear any accumulated mismatch in the reconstructed frames.

Table 2 summarizes the four types of redundancy that can be added to a P-MD video coder. First, the MD-PE encoder introduces redundancy, denoted ρ_a. Second, using a predictor that is less efficient than the SD predictor adds redundancy, denoted ρ_b. Third, the bit-rate ρ_c can be used solely to reduce mismatch. Finally, the redundancy ρ_d corresponds to any bit rate used to describe side information in excess of that used by an SD encoder. This includes extra bits used to describe motion vectors and any other MD parameters. This redundancy may be nontrivial if the total rate

Table 2
Summary of Redundancy Types

Type	Cause of redundancy	Symbol
(a)	Coding the prediction error signal(s) using MD	ρ_a
(b)	Using a predictor that is less efficient than the SD predictor	ρ_b
(c)	Sending an explicit signal for mismatch reduction	ρ_c
(d)	Sending side information	ρ_d

is low or if the encoder adapt its redundancy dynamically based on time-varying source and/or channel characteristics. The overall rate of a P-MD encoder can thus be expressed as

$$R = R^* + \rho = R^* + \rho_a + \rho_b + \rho_c + \rho_d. \tag{5}$$

C. Using P-MD Coders on a Packet-Loss Network

In a packet-loss network, compressed information is partitioned into multiple packets, and during transmission each packet has a (typically equal) probability of being lost p. Most real-world networks including the Internet and many wireless networks experience busty packet losses, i.e., many successive packets are often lost in a burst due to congestion or path failure. If the two descriptions from an MD coder are packetized such that each burst usually affects only one description, then at the burst level, the packet-loss network can be approximated as an ideal MD network. This factor has lead to the surge of interest in employing MD coders to transport signals over the Internet and wireless networks. This section describes the challenges associated with designing a P-MD coder and corresponding packetization schemes for operating over a packet-loss network.

In an ideal MD network, a description is either intact or completely lost. In a packet network, a description sent using multiple packets may be only partially lost. Further, losses may appear in both descriptions either at distinct times or simultaneously. In an ideal MD network, an MD encoder can easily anticipate possible decoding states exactly; however, in a packet-loss network this is challenging. If the two descriptions are partitioned into L packets, there are $2^L - 1$ possible sets the decoder might receive. Thus, mismatch control by the encoder is challenging.

As for the ideal MD network, two approaches to mismatch control over a packet-loss network are possible: mismatch coding or mismatch elimination. In practical MD video coders, mismatch coding by sending $2^L - 1$ mismatch signals is often abandoned, in favor of the simpler approach of sending an I-frame to clear up the error. Other approaches for mismatch coding include using a sync-frame [9] or SP-frame [18].

For mismatch elimination in a packet network, two approaches have been proposed. The first merges smart packetization with a mismatch-free P-MD encoder [10]. Motion-compensated prediction is constrained so that each description can be partitioned into multiple packets, each self-sufficient for prediction [10]. However, this requires latency as long as a group of pictures (GOP). In the second approach, a mismatch-free P-MD coder generates L descriptions, one for each packet. However, this may increase redundancy dramatically.

D. Optimization of a P-MD Coder

The performance of a P-MD coder in an MD or a packet network depends on the channel characteristics and the source statistics. Because these both vary with time, it is important that MD coders be able to be flexibly and dynamically adapt redundancy. The many sources of redundancy in a P-MD coder make this a challenging task.

For the ideal MD network, we need to allocate the total rate optimally among R^* and ρ, as described in (3). If the total rate is constrained within each frame period, then it is necessary to allocate the redundancy ρ to its individual components [see (5)]. Further, if the total rate can be allocated across multiple frames, then it is also important to optimally allocate rate and redundancy to the different frames. In a coder with mismatch, $D_{1,\text{mis}} \neq 0$ and typically increases over time. Thus, it is generally useful to allocate more redundancy to early frames to reduce the mismatch in subsequent frames.

For the packet-loss network, the goal is to minimize the overall distortion averaged over all possible packet loss scenarios and averaged over time, subject to a constraint on the overall rate. Except in the mismatch-free case (achievable with, e.g., the two approaches described in Section III-C), a packet loss of an early frame will affect the reconstruction of later frames. Choosing among the MD parameters $\mathcal{M}$ for each frame, taking into account current and future impact of loss, is a daunting task.

Two approaches are possible for the packet-loss network. The first chooses all MD parameters $\mathcal{M}$ across the GOP to minimize the average distortion, taking into account error propagation [19]. The second approach is recursive, using greedy optimization at each frame period (or smaller interval) to choose those MD parameters $\mathcal{M}$ that minimize the distortion for this frame, based on past frames [20]. The effect of errors on future frames is ignored.

When optimizing over either type of network, the coder should be able to flexibly adjust redundancy. As the channel or packet loss/failure parameter p becomes small, ideally an MD coder should add less redundancy so that it closely approximates an SD coder. As we will see in Section III, some proposed coders do not allow independent control over all types of redundancy. When one type of redundancy decreases, another necessarily increases.

III. Review of MD Video Coders

In this section, we describe specific implementations of MD video coding algorithms. We begin in Section III-A by describing MD coding algorithms useful for the MD-PE block of the P-MD encoders illustrated in Fig. 3. Sections III-B and III-C describe video coders which have no mismatch because they use a predictor in Class A. Then, Sections III-D and III-E describe video coders that use the SD predictor, both without and with explicit mismatch coding. Section III-F describes video coders that use a predictor in Class C, which trades prediction efficiency and mismatch, either with or without optional mismatch coding. Section III-G describes MD coders realized by applying unequal forward error correction (FEC) to different layers of a scalable bitstream. Depending on how many layers are included as references for prediction, these coders may have a predictor in Class A, B, or C.

Ideally, we would like to compare the different implementations of MD video coders in a fair and balanced way. Unfortunately, because of the variety of experimental results that are reported in the literature, this is difficult. The experiments use high- or low-motion test video sequences, different resolution formats, bit rates, frame rates, different error concealment methods, packetization schemes, and error patterns. Therefore, in this section, we purposely do not compare the performance of the different MD video coders. We refer the interested reader to the individual publications to learn more about comparative performance.

A. Multiple Description Algorithms

During recent years, a variety of practical MD compression algorithms have been proposed: subsampling either in the spatial, temporal, or frequency domain, MD quantization, and MD transform coding. These algorithms typically appear in the MD-PE encoder of the P-MD encoder of Fig. 3. Some of these methods were also recently reviewed in [6].

MD subsampling decomposes the original signal into subsets, either in the spatial, temporal, or frequency domain [21], [9], [22]–[24], [15], [8], [16], [11], [25], where each subset corresponds to a different description. These MD subsampling algorithms take advantage of the fact that spatially or temporally adjacent video data samples are correlated. Thus, one description can be estimated from the other. Representative algorithms include temporal frame interleaving [9], spatial pixel interleaving applied either to image samples [11], [22] or motion vectors [15], and transform coefficient interleaving [23], [24], [26]. Optimal partitioning strategies are considered in [25].

Without prefiltering, the redundancy and side distortion of MD subsampling algorithms are controlled by the source statistics. However, a well-designed prefilter can control the tradeoff between redundancy and side distortion [11], [22].

In MD quantization algorithms [5], [27]–[33], the output of a quantizer is assigned two indexes, one for each description. The MD decoder estimates the reconstructed signal based on the received index(es). The redundancy introduced by an MD scalar quantizer (MDSQ) and the corresponding side distortion are both controlled by the assignment of indexes to each quantization bin.

In MD transform coding (MDTC) [7], [34] a correlating transform introduces a controlled amount of redundancy between two sets of coefficients. Within each description, coefficients should be uncorrelated for maximum coding efficiency. At the decoder, missing coefficients can be estimated from the received description. For example, the pairwise correlating transform (PCT) [7] transforms a pair of random variables using a nonorthogonal linear transform. The two outputs each lie in one of two nonorthogonal subspaces. The optimal transform that achieves minimal side distortion for a given redundancy is parameterized by a single variable, which controls the amount of correlation introduced, which in turn controls the redundancy and side distortion.

B. MD Video Coders Using Multiple Class A Predictors

We consider here video coders that use multiple predictors to eliminate mismatch, starting with subsampling techniques [9], [8], [11] followed by MSDQ methods [10], [35].

In the video redundancy coding (VRC) algorithm [9], [36], a video sequence is temporally down-sampled into two subsets, essentially with every other frame in each subset. The frames in each subset are coded into a description using an SD video encoder. At the decoder, if only one description is received, the missing frames can be estimated from the received frames. In the context of the framework in Fig. 3, we see that the MD-PE encoder simply alternates each frame; hence, there is no type (a) redundancy. There is no mismatch in an ideal MD network, because the two predictors used at the encoder are exactly those of the side decoder. However, this comes at the expense of nonzero type (b) redundancy, which is introduced because of the decreased correlation between the alternated frames as compared to adjacent frames. For this coder, ρ_b is governed by the source and cannot be controlled by the encoder.

In a packet-loss network, VRC deals with uncertain losses by using sync-frames. In a packet-loss network, it is possible to adjust redundancy by adapting the frequency of sync-frames.

The general idea of using multiple states for prediction was proposed in [8] for a packet-loss network. The implementation of the multiple-state encoder in [8] was identical to VRC without sync-frames. State recovery for this coder was also presented in [37], where if one frame is lost, the correctly received frames are used to help recover the missing frame.

Instead of temporal subsampling, the independent flow md video coder (IF-MDVC) [11] uses the quincunx subsampling lattice on each video frame to generate two subsequences and applies a separate predictive encoder to each subsequence. Before down-sampling, each frame is low-pass filtered in the DCT domain using a prefilter. Adjusting the prefilter parameter controls both type (a) and type (b) redundancy.

One advantage of using MD subsampling algorithms to build P-MD vido coders is that because of the subsampling, no redundancy is needed for sending motion information. Therefore, ρ_d is close to zero. Another advantage is that it is easy to extend these algorithms to more than two descriptions by increasing the subsampling rate. However, it is expected that the redundancy ρ_b due to prediction inefficiency will increase rapidly as the number of descriptions grows.

Mismatch-free MD video coding using multistate prediction can also be accomplished using MDSQ [10], with mutually refining DPCM (MR-DPCM). Again, two predictors are used at the encoder to mimic the two side-decoder predictors. Here, the prediction errors are DCT transformed and quantized by an MDSQ. To ensure that the information received in both descriptions is mutually refining at the central decoder, a coarse quantizer is applied in the prediction loops. In MR-DPCM, redundancy is controlled by changing the index assignment of MDSQ. However, as ρ_a decreases, the prediction efficiency decreases so ρ_b increases.

Regunathan and Rose uses an estimation-theoretic approach to enhance MR-DPCM in [35]. They improve the quality of the reconstructed video displayed at the decoder in all instances by using an estimation algorithm that takes into account all information received by the decoder.

C. MD Video Coders Using One Class A Predictor

Next, we consider video coders that eliminate mismatch using a single predictor. To accomplish this, the predictor must form its prediction using information that is common to both descriptions. In general, this will incur type (b) redundancy because the predictor will be less efficient than an SD predictor.

One example is MD-SNR, which was presented in [12] as a reference for comparison. The MD-SNR coder is built using an H.263 SNR-scalable coder by duplicating the base layer (BL) into both descriptions and alternating the enhancement layer between the two descriptions. At the decoder, if only one description is received, only the BL is used for reconstruction. In H.263 SNR scalability, the prediction of the BL information uses only BL information. Therefore, MD-SNR introduces no mismatch. Redundancy in the MD-SNR codec is controlled by the size of the BL. Duplicating more information increases ρ_a but decreases ρ_b because the prediction efficiency is improved.

This prediction strategy was also used in [13], which codes the prediction error using a discrete wavelet transform (DWT). The bitstream corresponding to the coded bitplane of a block of DWT coefficients may be included in one, two, or more descriptions. The prediction is based only on information that is common to all descriptions.

D. MD Video Coders Using Class B Predictor Without Mismatch Coding

The Class B predictor is the SD predictor. As a result, all coders that use this predictor have $\rho_b = 0$. However, without some form of mismatch coding, these coders will all have mismatch $D_{1,\mathrm{mis}} \neq 0$. The mismatch can be controlled implicitly by adjusting the parameters of the MD-PE encoder, where more ρ_a reduces the mismatch. Alternatively, the mismatch can be coded with redundancy ρ_c to reduce or eliminate mismatch. This section describes the basic coders without mismatch coding.

The first coder we describe here is based on MDTC [12]. We call it MDTC-NMC, where NMC denotes no mismatch coding. A single prediction error is formed using an SD predictor with all information that would be available to the central decoder. The DCT is applied to the prediction error, followed by MDTC, where pairs of DCT coefficients have correlation introduced. One member of each pair is sent in each description. Motion vectors are duplicated in each description so that ρ_d is nonzero and depends on the spatial resolution. Type (a) redundancy ρ_a can be controlled by adjusting the transforms applied to each pair of coefficients. Larger ρ_a decreases both terms of the side distortion in (4). Total redundancy was allocated across time in [12] to reduce error propagation due to mismatch. However, in general, MDTC-NMC cannot completely eliminate mismatch. Therefore, periodic I-frames are inserted to clear up mismatch.

To avoid the type (d) redundancy associated with duplicating motion vectors, Kim and Lee proposed a coder that splits motion vectors and DCT coefficients of adjacent blocks into two descriptions using the quincunx lattice [15]. For this coder ρ_a, ρ_b, and ρ_d are all very small and fixed by the source, while $\rho_c = 0$. They use OBMC to improve the reconstruction of the side decoders; however, because of the minimal redundancy used, this coder may have significant side distortion.

In [14], Reibman *et al.* proposed the MD-split method, which uses the simplest possible MD algorithm for the MD-PE encoder: duplication or alternation. Using an H.263 bitstream, motion vectors are always duplicated, and a varying number of low-frequency coefficients are also duplicated. The number of coefficients to duplicate can be adapted easily based on varying source and channel statistics without explicitly informing the decoder. Duplicating more coefficients increases ρ_a and decreases both terms in the side distortion (4). This coder will always have mismatch unless all coefficients are duplicated.

In parallel, Comas *et al.* [38] proposed an algorithm with very similar properties. Again, coefficients are either duplicated or sent in just one of two descriptions. However, in [38], one description of a block contains all coefficients while the other contains only low-frequency coefficients. By alternating on a frame or block basis which description gets all coefficients and which gets a subset of coefficients, balanced distortion can be achieved for each description.

MD-split was optimized for an ideal MD network in [14] and for a packet-loss network in [20]. In each case, the effect of past mismatch was considered but the impact of future mismatch and error propagation was not considered.

The MD-Split coder duplicates the lower frequency coefficients only. Recently, Kim and Cho [39] extended this to allow any coefficient to be duplicated. Instead of operating on the DCT coefficients, the matching pursuits MD video coder (MP-MDVC) proposed by Tang *et al.* [40] duplicates and alternates matching pursuits atoms onto each description. Redundancy ρ_a is controlled by adjusting the number of duplicated atoms.

The drift compensated MD video coder (DC-MDVC) [11] also uses the SD predictor. Similar to IF-MDVC, it uses spatial down-sampling with prefiltering, but applied to the prediction error images instead of the original video. By adjusting the prefiltering, the redundancy ρ_a can be controlled. Both terms of the side distortion in (4) decrease as ρ_a increases.

E. Optional Mismatch Coding for Class B Predictors

All coders described in Section III-D may include optional mismatch coding, incurring redundancy ρ_c. One example of explicit mismatch coding is to use I-frames to reset errors.

An alternative is to code the mismatch signal [12], [40], [11], [41], [42]. These methods use all three predictors shown in Fig. 3, where $P_0(S_0)$ is the SD predictor and $P_i(S_i)$ mimics the predictor at each side decoder. For a class B predictor, the mismatch error signal is $E_{\mathrm{mis},i}(t) = P_0(S_0) - P_i(S_i)$. We note that for these coders it may not be possible to decompose the side distortion into the two terms as in (4).

Two mismatch coding strategies were proposed in [12], both using MDTC as the MD-PE encoder. We call them MDTC-FMC and MDTC-PMC, with the suffixes indicating full mismatch coding and partial mismatch coding, respectively. MDTC-FMC sends a mismatch correction signal $M_i(t) = X(t) - \breve{X}_i(t)$ on each description i using a quantizer Q_2 which is generally coarser than the quantizer used to send the prediction error $E_{p0}(t)$. Here, $\breve{X}_i$ indicates the reconstructed signal by decoder i without employing $M_i(t)$. Decoder i adds the quantized $M_i(t)$ to $\breve{X}_i$ to obtain a better side reconstruction that is free of mismatch. The side distortion is controlled by Q_2 only. By changing Q_2, ρ_c can be adjusted independently of ρ_a.

In MDTC, each description contains information in one of two nonorthogonal subspaces. Motivation for MDTC-PMC [12] arose by noting that most of the energy of the mismatch signal lies in the subspace orthogonal to its description. Therefore, MDTC-PMC codes $M_i(t)$ only in that orthogonal subspace. For the same Q_2, MDTC-PMC has lower redundancy ρ_c than MDTC-FMC but higher side distortion.

In [12], the decoder never uses the coded $M_i(t)$ when both descriptions are received. Tang *et al.* [40] use maximum likelihood estimation in MP-MDVC to improve the reconstruction quality of the central decoder by incorporating the information in coded $M_i(t)$. MP atoms to represent $M_i(t)$ are chosen to minimize the side distortion. The redundancy ρ_c is controlled by the number of atoms used to code $M_i(t)$. In [43], MP-MDVC was further optimized for a packet-loss network. A fast algorithm is presented to jointly select the number of atoms that are duplicated and the number of atoms used for mismatch coding, depending on the packet-loss conditions.

The mismatch coding described above codes the error signal $X(t) - \breve{X}_i(t)$. An alternate approach, applied to DC-MDVC, is to code the error signal $\tilde{X}_0(t) - \breve{X}_i(t)$ [11]. This will minimize the difference between $\tilde{X}_0(t)$ and $\tilde{X}_i(t)$, reducing the mismatch in following frames.

The coders above each send the mismatch error as an additional signal. Jagmohan and Ratakonda [41] send a single signal in each description which contains the sum of the SD prediction error and the mismatch, using MDTC. A similar idea was incorporated in an MDSQ coder by Lee *et al.* [42] and applied to video. In [42], the signal included in description i is $Q_i(X - P_0) + Q_i(P_0 - P_i), i = 1, 2$, where Q_i is the output of the MDSQ quantizer for description i. Both coders require that the same quantizer be applied to the prediction error and the mismatch signal. Thus, one cannot separately control the central and side distortion. Further, as the MD-PE parameters are changed to decrease ρ_a, ρ_c necessarily increases.

F. MD Video Coders Using Class C Predictors—With and Without Optional Mismatch Coding

In this section, we consider coders that use a predictor in Class C. These coders have very flexible redundancy allocation, enabling easy adaptation to varying channel conditions.

Motivated by the MD-DPCM speech coder [21], Wang and Lin [16] proposed a coder named MD motion compensation (MDMC). In MDMC, the central prediction forms a linear superposition of the past two reconstructed frames. The MD-PE encoder uses temporal subsampling similar to VRC, so that the side decoders will only receive every other frame. As a result, without optional mismatch coding, the side decoders will have mismatch. The weights for the linear superposition control both redundancy ρ_b and mismatch.

In [17], Kim *et al.* presented double-vector motion compensation (DMC), for use in a packet loss network. DMC also uses a linear superposition for motion compensation. Expressions for error propagation as a function of the superposition weights were presented. Optional mismatch coding was not considered.

The MDMC encoder [16] has also been designed to include mismatch coding. MDMC with mismatch coding has highly flexible redundancy control; the superposition weights control ρ_b, and the quantization parameter for coding mismatch controls both ρ_c and the side distortion. The coded mismatch signal enables not only the cancellation of the mismatch but also more accurate estimation of the missing description (i.e., estimating even frames from odd frames, and vice versa). MDMC operates slightly differently depending on the expected network. For ideal MD networks, the encoder stores the reconstructed past frames from the three possible decoding scenarios separately and forms $P_i(S_i)$ using past frames corresponding to different states S_i. For packet loss networks where it is difficult to anticipate possible decoding states, only the reconstructed frames from both descriptions are stored, and mismatch is controlled via both mismatch coding and periodic I-frames. Adaptation of the MDMC coder to channel loss conditions requires a good model that relates rate, redundancy, central distortion, and side distortion with encoder parameters. This is studied in [44].

G. MDC Through Unequal FEC

Instead of designing the source encoder to yield multiple descriptions directly, one can apply unequal cross-packet FEC to different parts of a scalable bitstream. This method, pioneered in [45] and [46], is commonly known as MD-FEC. As shown in Fig. 4, the original bitstream is partition into M layers, with layer k divided into k equal-length groups. An (M, k) Reed–Solomon (RS) code is then applied across k groups to yield M groups. Description k contains bits from group k from all layers. The decoder can recover first K layers of the original bitstream from any K descriptions by performing RS decoding. Redundancy and the associated side distortion are controlled by varying the layer partition $(R_1, R_2, \ldots, R_M)$. Given the probability of receiving K descriptions, $K = 1, 2, \ldots, M$, the optimal layer partition

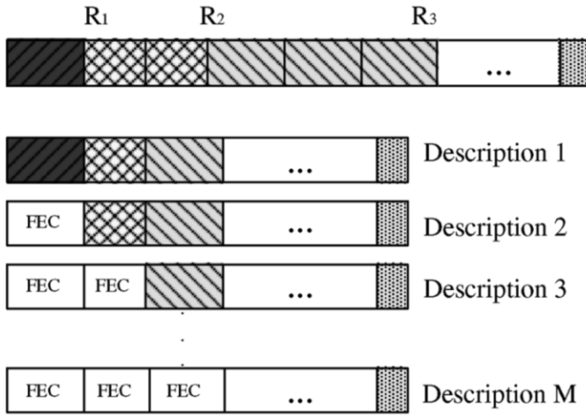


Fig. 4. MD-FEC algorithm.

$(R_1, R_2, \ldots, R_M)$ can be determined by minimizing the expected distortion subject to a total rate constraint (including channel coding) [47], [48].

MD-FEC has become a popular tool for generating MD video because it can be applied to any scalable video stream to generate an arbitrary number of descriptions. With a scalable video coder, each GoP is coded into an embedded stream. MD-FEC is applied to each GoP independently by partitioning the bitstream of each GoP into M layers and applying unequal FEC in different layers. Depending on how many layers are included as reference for temporal prediction, the coder can be considered as using either a Class A predictor (if only layer 1 is used for prediction), a Class B predictor without mismatch coding (when all layers are used for prediction), or a Class C predictor without mismatch coding (if $K, 1 < K < M$ layers are used).

H. Summary

Table 3 summarizes the P-MD video coding frameworks. Dynamic and flexible redundancy allocation is necessary to adapt to changing source and channel characteristics. In MD coders that eliminate mismatch completely, it is often difficult to flexibly allocate redundancy. For example, those coders that use one or more Class A predictors cannot achieve very low redundancy; as ρ_a decreases, prediction efficiency decreases and ρ_b increases. Thus, they may not be the best choice for networks with low loss rates.

A careful performance comparison among these coders is missing from the literature. In such a study, it would be essential to take into consideration factors such as optimal redundancy allocation, performance for high and low rates, in high- and low-motion sequences, in high- and low-loss networks, and robustness to changing network conditions. Ideally, to fully understand the impact of the MD parameters on performance, this comparison should use MD video coders built from a single structure for the SD video coder.

IV. Integration of Multiple Description Coding and Multiple Path Transport

The performance of MDC critically depends on whether the packets from different descriptions that correspond to the same or adjacent spatio-temporal segments in a video are likely to be lost simultaneously. One effective approach to reduce the likelihood for simultaneous loss is to send different descriptions through separate paths. We coin this transmission scheme multiple path transport (MPT) (a.k.a. "path diversity"). While each transport path may be unreliable, the chance that two paths simultaneously experience failures is typically low. MPT also enables load balancing in the network, thus reducing congestion and consequently packet losses. When a single path does not have sufficient bandwidth to carry an entire video stream, employing multiple paths can also effectively increase the aggregate end-to-end bandwidth.

Several research groups have explored the integration of MDC with MPT for video communications, both for video over wireless networks [49], [50] and over the Internet [8], [51], [52]. In video streaming applications, server diversity (a variant of path diversity) combined with either MDC or SDC has also been studied [53]–[57]. Instead of using multiple servers, the work in [58] relays the information from a server through separate peers using a peer-to-peer (P2P) networking architecture. These works have clearly shown the advantage of MPT over single-path transport (SPT) when the same MD or SD coder is used. Furthermore, combining MDC and MPT

Table 3
Summary of Frameworks for P-MD Video Coders

Multiple Class A predictors [9], [8], [11], [10]:
Multiple states, multiple predictors.
Mismatch elimination, $D_{1,mis} = 0$; $\rho_c = 0$.
Decreasing ρ_a increases ρ_b.
Single Class A predictor [12], [13]:
Single predictor based on information common to both descriptions.
Mismatch elimination, $D_{1,mis} = 0$; $\rho_c = 0$.
Decreasing ρ_a increases ρ_b.
Class B predictor, no mismatch coding [12], [20], [15]:
Single predictor identical to SD predictor.
No prediction inefficiency, $\rho_b = 0$.
Has mismatch, $D_{1,mis} \neq 0$, $\rho_c = 0$.
Class B predictor, with mismatch coding [12], [40], [11], [41], [42]:
SD predictor with two additional side predictions.
No prediction inefficiency, $\rho_b = 0$.
Mismatch coded with rate ρ_c to reduce or eliminate $D_{1,mis}$.
Class C predictor with optional mismatch coding [16], [17]:
Predictor that trades off prediction efficiency and mismatch.
Mismatch coded with optional rate ρ_c to eliminate mismatch.
Independent control of ρ_b and ρ_c.

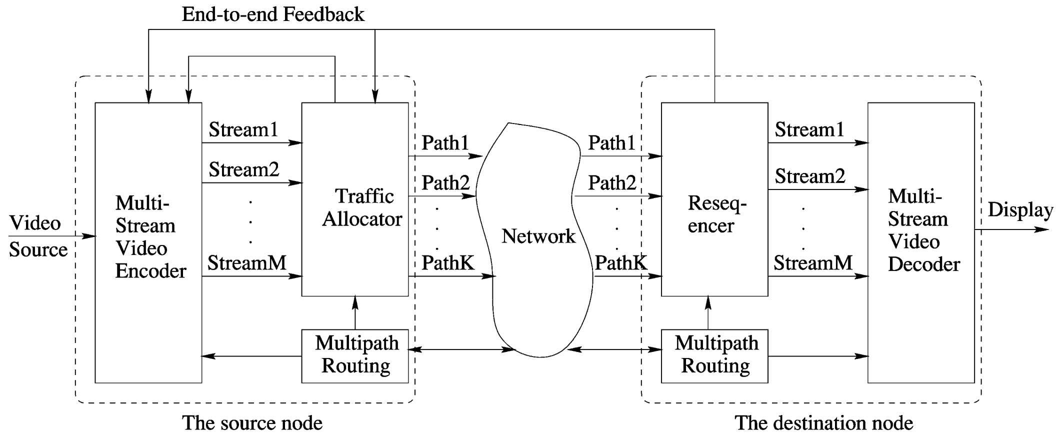


Fig. 5. General architecture of the system using multistream coding and multipath transport. Reprinted from [50]. Copyright © 2003 IEEE.

can lead to substantial performance gains over using SDC and SPT.

In Section IV-A, we discuss some of the fundamental issues involved in designing a system that integrates MDC with MPT. Then in Sections IV-B, IV-C, IV-D, we review three example systems that effectively integrate MDC with MPT for video transport over different networks.

A. *General System Architecture*

The general architecture of a system combining MDC with MPT is shown in Fig. 5. This figure assumes a multistream source coder with MD coders as a special case. In general, the number of paths and the available bandwidth and reliability on each path may change with time. We assume the system receives feedback about network quality of service (QoS) parameters (e.g., bandwidth, delay, and loss probabilities). At each update interval, a multipath routing layer discovers and sets up K paths between the source and destination. The transport layer continuously monitors path parameters and feeds back such information to the sender. Based on this information, the coder at the sender generates M ($M \neq K$ in general) multiple packetized bitstreams. (The feedback and interaction is only desirable, not necessary.) The packets from different substreams are distributed by the traffic allocator among K paths. At the receiver, the packets arriving from all the paths are put into a resequencing buffer, where they are reassembled into M substreams after a preset time-out period. Finally, the decoder reconstructs the transmitted video from the received packets in all the substreams.

Two important issues requiring close interaction between the source encoder and the transport layer are traffic allocation and path selection. For an MD source coder, traffic allocation should be such that packets from different descriptions that carry information about nearby spatio-temporal segments of the video sequence be spread over different paths to reduce the chance that these packets are simultaneously lost. Similarly, path selection should try to minimize the expected distortion of reconstructed video, which depends on the loss characteristics of individual paths as well as path correlation. Several papers have examined the effect of shared links among the chosen paths on the end-to-end video quality [59], [51], [60].

Table 4
OPNET simulation results using the MDMC coder

Transport Scheme	Average PSNR	Packet Loss Rate on Each Description
SPT	29.07 dB	6.3%
MPT	31.73 dB	3.0%

B. *Video Transport in Wireless Ad Hoc Networks*

Ad hoc networks are multihop wireless networks without a preinstalled infrastructure. MPT is very appealing for wireless *ad hoc* networks for several reasons: 1) a path can break down frequently because of node movements; 2) links are unreliable with frequent packet losses; and 3) individual links may not have adequate capacity to support high bandwidth service.

Mao *et al.* [50] proposed to combine multistream coding and MPT for video transport over *ad hoc* networks. Three multistream coding techniques with respective transport control protocols were compared: the MDMC video coder [16] (see also Section III-F) without retransmission or feed-back based adaptation, a H.263-based layered coder with constrained retransmission of the base-layer, and a feedback-based reference picture selection (RPS) scheme. These schemes were examined and compared both using Markov models at the link level and using the OPNET network modeler. The first two schemes were also implemented and evaluated on a testbed consisting of laptops capable of transmitting/receiving from two paths.

Table 4 summarizes the results from a simulated network using OPNET with 16 nodes moving in a 600×600 m region all at the same speed of 10 m/s. The video sequence (“Foreman,” QCIF) was coded using the MDMC coder at 118 kb/s. With SPT, all packets are transmitted over a single path, updated using the NIST dynamic source routing (DSR)

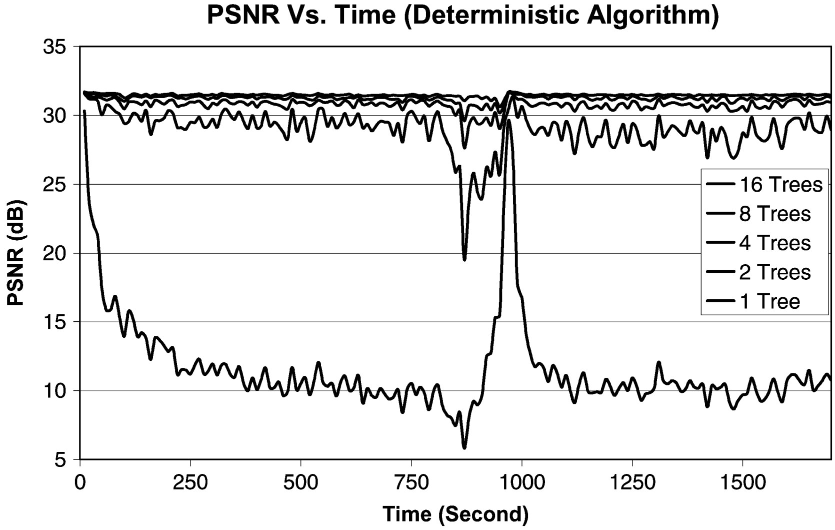


Fig. 6. Average PSNR versus time when the number of trees varied from 1 to 16, for sequence "Akiyo." Curves from top to bottom correspond to results obtained using 16, 8, 4, 2, and 1 distribution trees, respectively. Reprinted with permission from [58]. Copyright © 2003 Microsoft Corporation. All rights reserved.

protocol [61]. With MPT, the packets from the two descriptions are transported on two disjoint paths maintained by a multipath DSR (MDSR) protocol, which was developed by extending the DSR model. Table 4 shows that MPT leads to a lower packet loss rate on each description. The packet loss traces (see [50]) also show that the loss events on the two descriptions are less correlated with MPT. Both factors contribute to a substantial improvement in video quality.

C. Video Streaming in Content Delivery Networks

Apostolopoulos *et al.* [53] considered video streaming in a content delivery network (CDN) and examined the performance gains by coupling MDC and server diversity. In a traditional CDN, many servers are placed over the entire Internet, and when a client requests a media file, the closest server to the client is found and the requested file is streamed to the client from that server. To overcome the loss caused by server failures and packet losses in the transmission path, video is coded with multiple descriptions and stored on separate servers. Specifically, the two-state MD coder with temporal down-sampling [8] (see also Section III-B) is used. This MD-CDN scheme (which streams two descriptions from two separate servers) is compared to an SD-CDN approach, which streams a single stream (generated using a H.263 coder) from a single server.

There are, in general, three issues involved in designing a CDN: 1) where to place the servers in the network; 2) how to distribute the files over the servers; and 3) how to select the server(s) for a client request. The focus in [53] was on the performance gains achievable by varying strategies for 2) and 3) while employing three types of existing network infrastructure for server placement. Their simulation results show that MD-CDN can reduce the distortion of the received video significantly, and the gain is more significant when the servers are located at "hotspots" (e.g., locations of the data centers of an Internet service provider).

D. CoopNet Project

Padmanabhan *et al.* [58] explored video multicasting using the so-called CoopNet, which employs the combination of a dedicated server and many cooperating peers in a P2P network to distribute live or pre-encoded video. The rationale for using peers is to alleviate the overload at the server caused by a special event. A critical challenge in designing such a network is that the peer nodes are inherently unreliable. CoopNet addresses this issue by employing multiple distribution trees spanning the set of participating nodes. The streaming content is encoded using MDC and the descriptions are distributed over different trees. Each distribution tree may contain one or more descriptions.

The MD-FEC algorithm discussed in Section III-G is applied to each GoP of the scalable bitstream from an MPEG4-PFGS encoder [62] to generate MD video streams. Information about whether a description is received or not in each GoP is fed back periodically from nodes to the server through a reverse path of each tree. Based on this feedback, the probability $p(m)$ that m out of M descriptions are received is updated. Based on the estimated $p(m)$ and the rate-distortion curve of the PFGS video encoder, the server recalculates the optimal layer partition in each update interval.

The peer arrival and departure are modeled using the flash crowd traces recorded at the MSNBC site on September 11, 2001. It was assumed that clients stop forwarding traffic the moment they depart. Once a node in a tree departs, all its descendent nodes will not receive the descriptions contained in this tree until the next update interval. Thus, a description is either completely received or lost in a node, within each update interval. If it is lost in the current interval, it will be distributed from a different parent in the next interval.

Fig. 6 compares the PSNR curves obtained with different number of trees, when the number of descriptions is fixed at $M = 16$. This figure shows that increasing the number of distributing trees (equivalent to the number of paths K)

can increase PSNR significantly. The improvement is most significant when K increases from one to two. However, the performance saturates once K increases beyond 8. This result is consistent with those reported in [56], which compared the performance obtained by varying K when $M = 2$. They have found that when K increases from one to two, a significant gain can be observed. As K increases further, performance stays almost constant.

V. Conclusion

In this paper, we have considered various techniques for MD video coding. We focused on MD video coders that use temporal prediction and classified these coders based on their strategies for generating multiple descriptions, prediction, and mismatch control. Our goal has been to create a structure that clearly defines the competing factors that must be considered when designing an MD video coder. This structure has allowed us to describe the progress to date and lays the groundwork for future contributions in MD video coding. We also presented a general architecture for transporting MD video over multiple paths and showed through example systems that combining MDC with MPT can lead to substantial performance gains compared to using SDC and a single path.

Researchers interested in MDC have at their fingertips a wealth of promising directions. Coding structures that are likely to receive increasing attention are those using efficient prediction structures that eliminate mismatch, those using new Class C predictors that allow flexible tradeoffs between mismatch and redundancy, and those that can efficiently generate more than two descriptions. Optimizing MD video coders for use on packet-loss channels also merits additional attention, as does the application of distributed source coding [63] to MD video [64].

A fairly new research direction is to combine scalable coding with MDC to offer both rate scalability and error resilience. This can be achieved either by creating MD coders in which each description is scalable or by designing layered coders in which each layer is MD with a different amount of redundancy. The MD-FEC coder (Section III-G) achieves this through channel coding. Other works in this direction include [26], [65], and [66].

Finally, much work remains on integrating MD video with MPT, both in existing networking environments as well as in emerging communication infrastructures including peer-to-peer, cooperative communications (in which nodes assist each other to relay information), and sensor networks.

Acknowledgment

The authors would like to thank Y. Altunbasak, J. Apostolopoulos, A. Cengiz Begen, P. A. Chou, T. Nguyen, and A. Zakhor for providing information on their work relevant to this paper. The authors would also like to thank V. Vaishampayan and S. Mao for helpful technical discussions.

References

[1] J. K. Wolf *et al.*, "Source coding for multiple descriptions," *Bell Syst. Tech. J.*, vol. 59, pp. 1417–1426, Oct. 1980.

[2] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, pp. 1909–1921, Dec. 1980.

[3] A. A. El-Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Theory*, vol. 28, pp. 851–857, Nov. 1982.

[4] N. S. Jayant, "Subsampling of a DPCM speech channel to provide two 'self-contained' half-rate channels," *Bell Syst. Tech. J.*, vol. 60, no. 4, pp. 501–509, Apr. 1981.

[5] V. A. Vaishampayan, "Design of multiple description scalar quantizer," *IEEE Trans. Inform. Theory*, vol. 39, pp. 821–834, May 1993.

[6] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, pp. 74–93, Sep. 2001.

[7] Y. Wang *et al.*, "Multiple description coding using pairwise correlating transforms," *IEEE Trans. Image Processing*, vol. 10, pp. 351–366, Oct. 2001.

[8] J. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity," in *Proc. Visual Communications Image Processing*, 2001, pp. 392–409.

[9] S. Wenger, "Video redundancy coding in H.263+," presented at the Audio-Visual Services Over Packet Networks Conf., Aberdeen, U.K., 1997.

[10] V. Vaishampayan and S. John, "Balanced interframe multiple description video compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 1999, pp. 812–816.

[11] N. Franchi *et al.*, "Multiple video coding for scalable and robust transmission over IP," presented at the Packet Video Conf., Nantes, France, 2003.

[12] A. Reibman *et al.*, "Multiple description coding for video using motion compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 193–204, Mar. 2002.

[13] N. V. Boulgouris *et al.*, "Drift-free multiple description coding of video," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, 2001, pp. 105–110.

[14] A. Reibman *et al.*, "Multiple description video using rate-distortion splitting," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2001, pp. 978–981.

[15] C. Kim and S. Lee, "Multiple description coding of motion fields for robust video transmission," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 999–1010, Sep. 2001.

[16] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 438–453, Jun. 2002.

[17] C. Kim *et al.*, "Robust transmission of video sequence using double-vector motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 1011–1020, Sep. 2001.

[18] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 637–644, Jul. 2003.

[19] S. Lin and Y. Wang, "Analysis and improvement of multiple description motion compensation video coding for lossy packet networks," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 2002, pp. II-185–II-188.

[20] A. Reibman, "Optimizing multiple description video coders in a packet loss environment," presented at the Packet Video Conf. 2002, Pittsburgh, PA.

[21] A. Ingle and V. A. Vaishampayan, "DPCM system design for diversity systems with application to packetized speech," *IEEE Trans. Speech Audio Process.*, vol. 3, pp. 48–58, Jan. 1995.

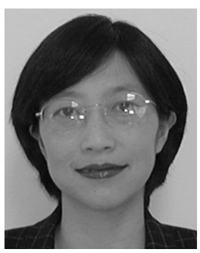
[22] Y. Wang and D. Chung, "Non-hierarchical signal decomposition and maximally smooth reconstruction for wireless video transmission," in *Mobile Multimedia Communications*, D. Goodman and D. Raychaudhuri, Eds. New York: Plenum, 1997, pp. 285–292.

[23] D. Chung and Y. Wang, "Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 895–908, Sep. 1999.

[24] ——, "Lapped orthogonal transforms designed for error resilient image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 752–764, Sep. 2002.

[25] I. V. Bajic and J. W. Woods, "Domain-based multiple description coding of images and video," *IEEE Trans. Image Process.*, vol. 12, pp. 1211–1225, Oct. 2003.

[26] S. Cho and W. A. Pearlman, "A full-featured, error resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees (SPIHT) algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 157–170, Mar. 2002.
[27] V. A. Vaishampayan and J. Domaszewicz, "Design of dentropy-constrained multiple description scalar quantizer," *IEEE Trans. Inform. Theory*, vol. 40, pp. 245–250, Jan. 1994.
[28] J.-C. Batllo and V. Vaishampayan, "Asymptotic performance of multiple description transform codes," *IEEE Trans. Inform. Theory*, vol. 43, pp. 703–707, Mar. 1997.
[29] M. Srinivasan and R. Chellappa *et al.*, "Multiple description subband coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 1998, pp. 684–688.
[30] S. D. Servetto *et al.*, "Multiple description lattice vector quantization," in *Proc. IEEE Data Compression Conf.*, 1998, pp. 13–22.
[31] H. Jafarkhani and V. Tarokh, "Multiple description trellis coded quantization," *IEEE Trans. Commun.*, vol. 47, pp. 799–803, Jun. 1999.
[32] M. Fleming and M. Effros, "Generalized multiple description vector quantization," in *Proc. IEEE Data Compression Conf.*, 1999, pp. 3–12.
[33] C. Tian and S. Hemami, "Universal multiple description scalar quantization: Analysis and design," in *Proc. IEEE Data Compression Conf.*, 2003, pp. 183–192.
[34] V. K. Goyal and J. Kovacevic, "Generalized multiple description coding with correlating transforms," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2199–2224, Sep. 2001.
[35] S. L. Regunathan and K. Rose, "Efficient prediction in multiple description video coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2000, pp. 1020–1023.
[36] S. Wenger *et al.*, "Error resilience support in H.263+," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 867–877, Nov. 1998.
[37] J. Apostolopoulos, "Error-resilient video compression through the use of multiple states," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2000, pp. 352–355.
[38] D. Comas *et al.*, "Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, 2001, pp. 581–586.
[39] I. K. Kim and N. I. Cho, "Error resilient video coding using optimal multiple description of DCT coefficients," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, 2003, pp. II-663–II-666.
[40] X. Tang and A. Zakhor, "Matching pursuits multiple description coding for wireless video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 566–575, Jun. 2002.
[41] A. Jagmohan and K. Ratakonda, "Multiple description coding of predictively encoded sequences," in *Proc. IEEE Data Compression Conf.*, 2002, pp. 13–22.
[42] Y. Lee *et al.*, "A drift-free motion-compensated predictive encoding technique for multiple description coding," in *Proc. IEEE Int. Multimedia Expo*, vol. 3, 2003, pp. III-581–III-584.
[43] T. Nguyen and A. Zakhor, "Matching pursuits based multiple description video coding for lossy environments," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2003, pp. I-57–I-60.
[44] S. Lin *et al.*, "Rate-distortion analysis of the multiple description motion compensation video coding scheme," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, 2003, pp. III-401–III-404.
[45] A. Albanese *et al.*, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, pp. 1737–1744, Nov. 1996.
[46] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," Proc. SPIE, Conf. Wavelet Applicat. Digital Image Process. XIX, vol. 2847, pp. 376–387, 1996.
[47] A. E. Mohr *et al.*, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 819–829, Jun. 2000.
[48] R. Puri and K. Ramchandran, "Multiple description source coding through forward error correction codes," in *Proc. 33rd Asilomar Conf. Signals, System Comp.*, vol. 1, 1999, pp. 342–346.
[49] N. Gogate *et al.*, "Supporting video/image applications in a mobile multihop radio environment using route diversity and multiple description coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 777–792, Sep. 2002.
[50] S. Mao *et al.*, "Video transport over ad hoc networks: Multistream coding with multipath transport," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1721–1737, Dec. 2003.
[51] A. Begen *et al.*, "Multi-path selection for multiple description encoded video streaming," in *Proc. IEEE Int. Conf. Commun.*, vol. 3, 2003, pp. 1583–1589.
[52] J. Chakareski *et al.*, "Video streaming with diversity," in *IEEE Int. Multimedia Expo*, vol. 1, 2003, pp. 9–12.
[53] J. Apostolopoulos *et al.*, "On multiple description streaming in content delivery networks," in *Proc. IEEE INFOCOM*, 2002, pp. 1736–1745.
[54] T. Nguyen and A. Zakhor, "Protocols for distributed video streaming," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2002, pp. III-185–III-188.
[55] A. Majumdar *et al.*, "Distributed multimedia transmission from multiple servers," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2002, pp. III-177–III-180.
[56] J. Chakareski and B. Girod, "Server diversity in rate-distortion optimized media streaming," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2003, pp. III-645–III-648.
[57] A. Begen *et al.*, "Rate-distortion optimized on-demand media streaming with server diversity," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2003, pp. III-657–III-660.
[58] V. Padmanabhan *et al.*. (2003, Mar.) Resilient peer-to-peer streaming. Microsoft Res.. [Online]. Available: http://www.research.microsoft.com/projects/CoopNet/papers/msr-tr-2003-11.pdf
[59] J. Apostolopoulos *et al.*, "Modeling path diversity for multiple description video communication," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, 2002, pp. III-2161–III-2164.
[60] T. Nguyen and A. Zakhor, "Path diversity with forward error correction (pdf) system for packet switched networks," in *Proc. IEEE INFOCOM*, vol. 1, 2003, pp. 663–672.
[61] J. Johnson *et al.*, The dynamic source routing protocol for mobile *ad hoc* networks, in IETF Internet draft (draft-ietf-manet-dsr-03.txt), Oct. 1999.
[62] F. Wu *et al.*, "A framework for efficient progressive fine granularity scalability video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 332–344, Mar. 2001.
[63] B. Girod *et al.*, Distributed video coding, in *Proc. IEEE*, 93, pp. 71–83, Jan. 2005.
[64] A. Jagmohan *et al.*, "WYZE-PMD based multiple description video codec," in *IEEE Int. Multimedia Expo*, vol. 1, 2003, pp. I-569–I-572.
[65] P. Chou *et al.*, "Layered multiple description coding," presented at the Packet Video Conf., Nantes, France, 2003.
[66] H. Wang and A. Ortega, "Robust video communication by combining scalability and multiple description coding techniques," *Proc. SPIE, Image Video Commun. Process.*, vol. 5022, pp. 111–124, May 2003.



Yao Wang (Fellow, IEEE) received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree from the University of California, Santa Barbara, in 1990, all in electrical engineering.

Since 1990, she has been on the faculty of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY. She is the leading author of the textbook *Video Processing and Communications* and has published over 150 papers in journals and conference proceedings. Her research interests include video communications and multimedia signal processing.

Dr. Wang received the New York City Mayor's Award for Excellence in Science and Technology in the Young Investigator Category in year 2000. She was elected an IEEE Fellow for contribution to video processing and communications in 2004. She is the Co-Winner of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2003.

Amy R. Reibman (Fellow, IEEE) received the Ph.D. in electrical engineering from Duke University, Durham, NC, in 1987.

From 1988 to 1991, she taught electrical engineering at Princeton University, Princeton, NJ. In 1991, she joined AT&T and is currently a Technical Consultant in the Communication Sciences Research Department at AT&T Lab.-Research, Florham Park, NJ. Her research interests include video compression systems for transport over packet and wireless networks and video quality metrics.

Dr. Reibman won the IEEE Communications Society Leonard G. Abraham Prize Paper Award in 1998. She was the Technical Co-Chair of the IEEE International Conference on Image Processing in 2002. She was elected an IEEE Fellow for contributions to the transport of video over networks.

Shunan Lin (Member, IEEE) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1995, the M.S. degree from the Institute of Automation, Chinese Academy of Sciences Beijing, China, in 1998, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 2004, all in electrical engineering.

Since 2004, he has been with Harmonic Inc, White Plains, NY. Between 2000 and 2003, he interned at Microsoft Research, Beijing, China, and Mitsubishi Electric Research Laboratory, Murray Hill, NJ. His research interests include the field of video coding, processing, and transmission.

Dr. Lin was the Co-Winner of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2003.