Rate-Distortion Optimization for Video Compression

Gary J. Sullivan, PictureTel Corporation
100 Minuteman Rd., Andover, MA 01810 USA
Tel: +1 978 623 4324, Fax: +1 978 749 2804
garys@pictel.com

Thomas Wiegand, University of Erlangen-Nuremberg
Cauerstraße 7/NT, D-91052 Erlangen, Germany
Tel: +49 9131 85 7117, Fax: +49 9131 85 8849
wiegand@nt.e-technik.uni-erlangen.de

ITU-T SG16 Document Q15-E-26
Draft for submission to IEEE Signal Proc. Magazine, Nov. 1998 issue

July 15, 1998

1 Introduction

Motion video data consists essentially of a time-ordered sequence of pictures, and cameras typically generate approximately 24, 25, or 30 pictures (or *frames*) per second. This results in a large amount of data, and therefore demands the use of compression. For example assume that each picture has a relatively low "QCIF" resolution, i.e., 176 × 144 samples, that each sample is digitally represented with 8 bits, and that we skip two out of every three pictures in order to cut down the bit rate. For color pictures, three color component samples are necessary to represent a sufficient color space for each pixel. In order to transmit even this relatively low-fidelity sequence of pictures, the raw source data rate is still more than 6 Mbit/s. However, today's low-cost transmission channels often operate at much lower data rates so that the data rate of the video signal needs to be further compressed. For instance, using V.34 modems that transmit at most 33.4 Kbit/s over dial-up analog phone lines, we still need to compress the video bit rate further by a factor of about 200 (more if audio is consuming 6 Kbit/s of that same channel or if the phone line is too noisy for achieving the full bit rate of V.34).

One way of compressing video content is simply to compress each picture, using an image coding syntax such as JPEG [1]. The most common "baseline" JPEG scheme consists of breaking up the image into equal size blocks. These blocks are transformed by a discrete cosine transform (DCT), and the DCT coefficients are then quantized and transmitted using variable length codes. We will refer to this kind of coding scheme as INTRA-frame coding, since the picture is coded without referring to other pictures in the video sequence. In fact, such INTRA coding alone is in common use as a video coding method today in production-quality editing systems which demand rapid access to any frame of video content.

However, improved compression performance can be attained by taking advantage of the large amount of temporal redundancy in video content. We will refer to such techniques as INTER-frame coding. Usually, much of the depicted scene is essentially just repeated in picture after picture without any significant change. It should be obvious then that the video can be represented more efficiently by coding only the changes in the video content, rather than coding each entire picture repeatedly. This ability to use the temporal domain redundancy to improve coding efficiency is what fundamentally distinguishes video compression from still image compression.

A simple method of improving compression by coding only the changes in a video scene is called conditional replenishment (CR), and it was the only temporal redundancy reduction method used in the first digital video coding standard, ITU-T Rec. H.120 [2]. CR coding consists of sending signals to indicate which areas of a picture can just be repeated, and sending new coded information to replace the changed areas. CR thus allows a choice between one of two modes of representation for each area, which are called the SKIP mode and the INTRA mode. However, CR coding has a significant shortcoming, which is its inability to refine an approximation. Often the content of an area of a prior picture can be a good approximation of the new picture, needing only a minor alteration to become a better representation. But CR coding allows only exact repetition or complete replacement of each picture area. Adding a third type of "prediction mode," in which a refining *frame difference* approximation can be sent, results in a further improvement of compression performance.

The concept of frame difference refinement can also be taken a step further, by adding *motion-compensated* prediction (MCP). Most changes in video content are typically due to the motion of objects in the depicted scene relative to the imaging plane, and a small amount of motion can result in a large difference in the values of the pixels in a picture area, especially near the edges of an object. Often, *displacing* an area of the prior picture by a few pixels in spatial location can result in a significant reduction in the amount of information that needs to be sent as a frame difference approximation. This use of spatial displacement to form an approximation is known as motion compensation and the encoder's search for the best spatial displacement approximation to use is known as motion estimation. The coding of the resulting difference signal for the refinement of the MCP signal is known as displaced frame difference (DFD) coding.

Hence, the most successful class of video compression designs are called hybrid codecs. The naming of this coder is due to its construction as a hybrid of motion handling and picture coding techniques, and the term codec is used to refer to both the coder and decoder of a video compression system. Figure 1 shows such a *hybrid* coder. Its design and operation involves the optimization of a number of decisions, including

1. How to segment each picture into areas,
2. Whether or not to replace each area of the picture with completely new INTRA-picture content,
3. If not replacing an area with new INTRA content
   (a) How to do motion estimation, i.e, how to select the spatial shifting displacement to use for INTER-picture predictive coding (with a zero-valued displacement being an important special case),
   (b) How to do DFD coding, i.e., how to select the approximation to use as a refinement of the INTER prediction (with a zero-valued approximation being an important special case), and
4. If replacing an area with new INTRA content, what approximation to send as the replacement content.

At this point, we have introduced a problem for the engineer who designs such a video coding system, which is: *What part of the image should be coded using what method*? If the possible modes of operation are restricted to INTRA coding and SKIP, the choice is relatively simple. However, hybrid video codecs achieve their compression performance by employing several modes of operation that are adaptively assigned to parts of the encoded picture, and there is a dependency between the effects of the two motion estimation and DFD coding stages of INTER coding. The modes of operation are in general associated with signal dependent rate-distortion characteristics, and rate-distortion trade-offs are inherent in the design of each of these aspects. The second and third items in particular are unique to motion video coding. The optimization of these decisions in the design and operation of a video coder is the primary topic of this article. Some further techniques which go somewhat beyond this model will also be discussed.

[Begin Sidebar Inset Article #1]
A History of Existing Visual Coding Standards

H.120: The first international digital video coding standard [2]. It may have even been the first international digital compression standard for natural continuous-tone visual content of any kind (whether video or still picture). H.120 was developed by the ITU-T organization (then called the CCITT), and received final approval in 1984. It was then revised in 1988. It originally was a conditional replenishment (CR) coder with differential pulse-code modulation (DPCM), scalar quantization, and variable length coding, and had an ability to switch to quincunx sub-sampling for bit-rate control. In 1988, a second version of H.120 added motion compensation and background prediction. (None of the later completed standards have yet included background prediction again, although a form of it is in the draft of the future MPEG-4 standard.) Its operational bit rates were 1544 and 2048 Kbit/s. H.120 is essentially no longer in use today, although a few H.120 systems are rumored to still be in operational condition.

H.261: The first widespread practical success – a video codec capable of operation at affordable telecom bit rates (typically 80-320 Kbit/s) [3]. It was the first standard to use the basic typical structure we find still predominant today (16×16 macroblock motion compensation, 8×8 block DCT, scalar quantization, and two-dimensional run-level variable-length entropy coding). H.261 was approved by the ITU-T in early 1991 (with technical content completed in late 1990). It was later revised in 1993 to include a backward-compatible high-resolution graphics transfer mode. Its target bit rate range was 64-2048 Kbit/s.

JPEG: A highly successful continuous-tone still picture coding standard named after the Joint Photographic Experts Group which developed it [1]. Anyone who has browsed the World-Wide Web has experienced JPEG. JPEG (IS 10918-1 / ITU-T T.81) was originally approved in 1992, and was developed as an official joint project of both the ISO/IEC JTC1 and ITU-T organizations. In its typical use, it is essentially H.261 Intra coding with prediction of average values and an ability to customize the quantizer reconstruction scaling and the entropy coding to the specific picture content. However, there is much more in the JPEG standard than what is typically described or used. In particular, this includes progressive coding, lossless coding, and arithmetic coding.

MPEG-1: A widely successful video codec capable of approximately VHS videotape quality or better at about 1.5 Mbit/s and covering a bit rate range of about 1-2 Mbit/s [4]. MPEG-1 gets its acronym from the Moving Pictures Experts Group which developed it [4]. MPEG-1 video (IS 11172-2) was a project of the ISO/IEC JTC1 organization and was approved in 1993. In terms of technical features, it added bi-directionally predicted frames (known as B-frames) and half-pixel motion. (Half-pixel motion had been proposed during the development of H.261 but was thought to be too complex at the time.) It provided superior quality than H.261 when operated at higher bit rates. (At bit rates below perhaps 1 Mbit/s, H.261 performs better, as MPEG-1 was not designed to be capable of operation in this range.)

MPEG-2: A step higher in bit rate, picture quality, and popularity. MPEG-2 forms the heart of broadcast-quality digital television for both standard-definition and high-definition television (SDTV and HDTV) [7]. MPEG-2 video (IS 13818-2 / ITU-T H.262) was designed to encompass MPEG-1 and to also provide high quality with interlaced video sources at much higher bit rates. Although usually thought of as an ISO standard, MPEG-2 video was developed as an official joint project of both the ISO/IEC JTC1 and ITU-T organizations, and was completed in late 1994. Its primary new technical features were efficient handling of interlaced-scan pictures and hierarchical bit usage scalability. Its target bit rate range was approximately 4-30 Mbit/s.

H.263: The first codec designed specifically to handle very low bit rate video, and its performance in that arena is still state-of-the-art [5]. H.263 is the current best standard for practical video telecommunication. Its original target bit rate range was about 10-30 Kbit/s, but this was broadened during development to perhaps 10-2048 Kbit/s as it became apparent that it could be superior to H.261 at any bit rate. H.263 (version 1) was a project of the ITU-T and was approved in early 1996 (with technical content completed in late 1995). The key new technical features of H.263 were variable block-size motion compensation, overlapped block motion compensation, picture-extrapolating motion vectors, three-dimensional run-level-last variable-length coding, median MV prediction, and more efficient overhead signaling (and, relative to H.261, arithmetic coding, half-pixel motion, and bi-directional prediction – but the first of these three features was also found in JPEG and the other two were in MPEG-1). At very low bit rates (e.g., below 30 Kbit/s), H.263 can code with the same quality as H.261 using half or less than half the bit rate [6]. At greater bit rates (e.g., above 80 Kbit/s) it can provide a more moderate degree of performance superiority over H.261. (See also H.263+ below.)

H.263+: Technically a second version of H.263 [5]. The H.263+ project added a number of new optional features to H.263. One notable technical advance over prior standards is that H.263 version 2 was the first video coding standard to offer a high degree of error resilience for wireless or packet-based transport networks. H.263+ also added some improvements in compression efficiency, custom and flexible video formats, scalability, and backward-compatible supplemental enhancement information. It was approved in January of 1998 by the ITU-T (with technical content completed in September 1997). It extends the effective bit rate range of H.263 to essentially any bit rate and any progressive-scan (non-interlace) picture formats and frame rates, and H.263+ is capable of superior performance relative to any existing standard over this entire range. The first author was the editor of H.263 during the H.263+ project and is the Rapporteur (chairman) of the ITU-T Advanced Video Coding Experts Group (SG16 Q15) which developed it.

[End Sidebar Inset Article #1]

[Begin Sidebar Inset Article #2]
An Overview of Future Visual Coding Standardization Projects

MPEG-4: A future visual coding standard for both still and moving visual content. The ISO/IEC SC29 WG11 organization is currently developing two drafts, called version 1 and version 2 of MPEG-4 visual. Final approval of version 1 is planned in January 1999 (with technical content completed in October 1998), and approval of version 2 is currently planned for approximately one year later. MPEG-4 visual (which will become IS 14496-2) will include most technical features of the prior video and still-picture coding standards, and will also include a number of new features such as zero-tree wavelet coding of still pictures, segmented shape coding of objects, and coding of hybrids of synthetic and natural video content. It will cover essentially all bit rates, picture formats, and frame rates, including both interlaced and progressive-scan video pictures. Its efficiency for predictive coding of normal camera-view video content will be essentially similar to that of H.263 for non-interlaced video sources and essentially similar to that of MPEG-2 for interlaced sources. For some special purpose and artificially generated scenes, it will provide significantly superior compression performance and new object-oriented capabilities. It will also contain a still image coder which has improved compression quality relative to JPEG at low bit rates.

H.263++: Future enhancements of H.263. The H.263++ project is considering adding more optional enhancements to H.263, and is currently scheduled for completion late in the year 2000. It is a project of the ITU-T Advanced Video Coding Experts Group (SG16 Q15).

JPEG-2000: A future new still-picture coding standard. JPEG-2000 is a joint project of the ITU-T SG8 and ISO/IEC JTC1 SC29 WG1 organizations. It is scheduled for completion late in the year 2000.

H.26L: A future new generation of video coding standard with improved efficiency, error resilience, and streaming support. H.26L is currently scheduled for approval in 2002. It is a project of the ITU-T Advanced Video Coding Experts Group (SG16 Q15).

[End Sidebar Inset Article #2]

[Begin Sidebar Inset Article #3]
Standard Hybrid Video Codec Terminology

prediction mode: A method which is selected for use in approximating a picture region.

block: A rectangular region (normally of size 8 × 8) in a picture. The Discrete Cosine Transform (DCT) in standard video coders always operates on 8 × 8 block regions.

macroblock: A region of size 16 × 16 in the luminance picture which is associated with a prediction mode.

motion vector (MV): A spatial displacement offset for use in the prediction of an image region. In the INTER prediction mode a MV affects a macroblock region, while in the INTER+4V prediction mode, an individual MV is sent for each of the four 8 × 8 blocks in a macroblock.

INTRA **mode:** A prediction mode in which the picture content of a macroblock region is represented without reference to a region in any previously decoded picture.

SKIP **mode:** A prediction mode in which the picture content of a macroblock region is represented as a copy of the macroblock in the same location in a previously decoded picture.

INTER **mode:** A prediction mode in which the picture content of a macroblock region is represented as the sum of a motion-compensated prediction using a motion vector, plus (optionally) a decoded residual difference signal representation.

INTER+4V **mode:** A prediction mode in which the picture content of a macroblock region is represented as in the INTER mode, but using four motion vectors (one for each 8 × 8 block in the macroblock).

INTER+Q **mode:** A prediction mode in which the picture content of a macroblock is represented as in the INTER mode, and a change is indicated for the inverse quantization scaling of the decoded residual signal representation.

[End Sidebar Inset Article #3]

2 Motion-Compensated Video Coding Analysis

Consider the nth coded picture of size $W \times H$ in a video sequence, consisting of an array $\boldsymbol{I}_n(\boldsymbol{s})$ of color component values, e.g., $Y_n(\boldsymbol{s}), Cb_n(\boldsymbol{s})$, and $Cr_n(\boldsymbol{s})$, for each pixel location $\boldsymbol{s} = (x, y)$, in which x and y are integers such that $0 \leq x < W$ and $0 \leq y < H$. The decoded approximation of this picture will be denoted as $\widetilde{\boldsymbol{I}}_n(\boldsymbol{s})$.

The typical video decoder (see Figure 1) receives a representation of the picture which is segmented into some number K of distinct regional areas $\{\boldsymbol{\mathcal{A}}_{i,n}\}_{i=1}^{K}$. For each area, a prediction mode signal $p_{i,n} \in \{0, 1\}$ is received indicating whether or not the area is predicted from the prior picture. For the areas that are predicted from the prior picture, a motion vector (MV), denoted $\boldsymbol{v}_{i,n}$ is received. The MV specifies a spatial displacement for motion compensation of that region. Using the prediction mode and MV, a motion-compensated prediction $\widehat{\boldsymbol{I}}_n(\boldsymbol{s})$ is formed for each pixel location $\boldsymbol{s} \in \boldsymbol{\mathcal{A}}_{i,n}$

$$\widehat{\boldsymbol{I}}_n(\boldsymbol{s}) = p_{i,n} \cdot \widetilde{\boldsymbol{I}}_{n-1}(\boldsymbol{s} - \boldsymbol{v}_{i,n}) \ , \ \boldsymbol{s} \in \boldsymbol{\mathcal{A}}_{i,n} \ . \tag{1}$$

(Note: The motion vector $\boldsymbol{v}_{i,n}$ has no effect if $p_{i,n} = 0$ and so the MV is therefore normally not sent in that case.)

In addition to the prediction mode and MV information, the decoder receives an approximation $\widetilde{\boldsymbol{R}}_{i,n}(\boldsymbol{s})$ of the DFD residual error $\boldsymbol{R}_{i,n}(\boldsymbol{s})$ between the true image value $\boldsymbol{I}_n(\boldsymbol{s})$ and its motion-compensated prediction $\widehat{\boldsymbol{I}}_n(\boldsymbol{s})$. It then adds the residual signal to the prediction to form the final coded representation

$$\widetilde{\boldsymbol{I}}_n(\boldsymbol{s}) = \widehat{\boldsymbol{I}}_n(\boldsymbol{s}) + \widetilde{\boldsymbol{R}}_{i,n}(\boldsymbol{s}) \ , \ \boldsymbol{s} \in \boldsymbol{\mathcal{A}}_{i,n} \ . \tag{2}$$

Since there is often no movement in large parts of the picture, and since the representation of such regions in the previous picture may be adequate, video coders often provide special provisions for a SKIP mode of area treatment which is efficiently transmitted using very short code words ($p_{i,n} = 1, \boldsymbol{v}_{i,n} = \boldsymbol{0}, \widetilde{\boldsymbol{R}}_{i,n}(\boldsymbol{s}) = \boldsymbol{0}$).

In video coders designed primarily for natural camera-view scene content, often little real freedom is given to the encoder for choosing the segmentation of the picture into region areas. Instead, the segmentation is typically either fixed to always consist of a particular-size two-dimensional block size (typically 16 × 16 pixels for prediction mode signals and 8 × 8 for DFD residual content) or in some cases it is allowed to switch adaptively between block sizes (such as allowing the segmentation used for motion compensation to have either a 16 × 16 or 8×8 block size). This is because providing the encoder more freedom to specify a precise segmentation has generally not yet resulted in a significant improvement of compression performance for natural camera-view scene content (due to the number of bits needed to specify the segmentation), and also because determining the best possible segmentation in an encoder can be very complex. However, in special applications (especially those including artificially-constructed picture content rather than camera-view scenes), segmented object-based coding may be justified. The optimization of coders that use object segmentation is discussed in an accompanying article [8].

2.1 Distortion Measures

Rate-distortion optimization requires an ability to measure distortion. However, the perceived distortion in visual content is a very difficult quantity to measure, as the characteristics of the human visual system are complex and not well understood. This problem is aggravated in video coding, because the addition of the temporal domain relative to still-picture coding further complicates the issue. In practice, highly imperfect distortion models such as sum of squared differences (SSD) or its equivalents known as mean squared error (MSE) or peak signal-to-noise ratio (PSNR) are used in most actual comparisons. They are defined by

$$\mathrm{SSD}_{\boldsymbol{\mathcal{A}}}(F,G) = \sum_{\boldsymbol{s}\in\boldsymbol{\mathcal{A}}} |F(\boldsymbol{s}) - G(\boldsymbol{s})|^2 \tag{3}$$

$$\mathrm{MSE}_{\boldsymbol{\mathcal{A}}}(F,G) = \frac{1}{|\boldsymbol{\mathcal{A}}|}\mathrm{SSD}_{\boldsymbol{\mathcal{A}}}(F,G) \tag{4}$$

$$\mathrm{PSNR}_{\boldsymbol{\mathcal{A}}}(F,G) = 10\log_{10}\frac{(255)^2}{\mathrm{MSE}_{\boldsymbol{\mathcal{A}}}(F,G)} \text{ decibels} \tag{5}$$

Another distortion measure in common use (since it is often easier to compute) is the sum of absolute differences (SAD)

$$\mathrm{SAD}_{\boldsymbol{\mathcal{A}}}(F,G) = \sum_{\boldsymbol{s}\in\boldsymbol{\mathcal{A}}} |F(\boldsymbol{s}) - G(\boldsymbol{s})| \tag{6}$$

These measures are often applied to only the luminance field of the picture during optimization processes, but better performance can be obtained by including all three color components. (The chrominance components are often treated as something of a minor nuisance in video coding; since they need only about 10% of the bit rate of the luminance they provide a limited opportunity for optimization gain.)

[Begin Sidebar Inset Article #4] Complicating Factors in Video Coding Optimization

The video coder model described in this article is useful for illustration purposes, but in practice actual video coder designs often differ from it in various ways that complicate design and analysis. Some of the important differences are described in the following few paragraphs.

Color chrominance components (e.g., $Cb_n(\boldsymbol{s})$ and $Cr_n(\boldsymbol{s})$) are often represented with lower resolution (e.g., $\frac{W}{2} \times \frac{H}{2}$) than the luminance component of the image $Y(\boldsymbol{s})$. This is because the human psycho-visual system is much more sensitive to brightness than to chrominance, allowing bit rate savings by coding the chrominance at lower resolution. In such a system, the method of operation must be adjusted to account for the difference in resolution (for example, by dividing the MV values by two for chrominance components).

Since image values $\boldsymbol{I}_n(\boldsymbol{s})$ are defined only for integer pixel locations $\boldsymbol{s} = (x, y)$ within the rectangular picture area, the above model will work properly in the strict sense only if every motion vector $\boldsymbol{v}_{i,n}$ is restricted to have an integer value and only a value which causes access to locations in the prior picture which are within the picture's rectangular boundary. These restrictions, which are maintained in some early video coding methods such as ITU-T Rec. H.261 [3], are detrimental to performance. More recent designs such as ITU-T Rec. H.263 [5] support the removal of these restrictions, by using interpolation of the prior picture for any fractional-valued MVs (normally half-integer values) and extrapolation outside the boundaries of the prior picture. The prediction of an image area may also be filtered to avoid high frequency artifacts (as in Rec. H.261 [3]).

Often there are interactions between the coding of different regions in a video coder. The number of bits needed to specify a MV value may depend on the values of the MVs in neighboring regions. The areas of influence of different MVs can be overlapping [9, 10, 11], and the areas of influence of residual difference signals can also be overlapping. While these cross-dependencies can improve coding performance, they can also complicate the task of optimizing the decisions made in an encoder. For this reason these cross-dependencies are often neglected (or only partially accounted for) during encoder optimization.

One important and often-neglected interaction between the coding of video regions is the temporal propagation of error. The fidelity of each area of a particular picture will affect the ability to use that picture area for the prediction of subsequent pictures. Real-time encoders must neglect this aspect to a large extent, since they cannot tolerate the delay necessary for optimizing a temporal sequence of decisions with accounting for the temporal effects on multiple pictures. However, even non-real-time encoders also often neglect to account for this propagation in any significant way, due to the sheer complexity of adding this extra dimension to the analysis. An example for the exploitation of temporal dependencies in video coding can be found in [12]. The work of Ramchandran, Ortega, and Vetterli in [12] was extended by Lee and Dickinson in [13].

[End of Sidebar Inset Article #4]

2.2 Effectiveness of Basic Technical Features

In the previous sections we described the various technical features of a basic modern video coder. The effectiveness of these features and the dependence of this effectiveness on video content is shown in Figure 2. The upper plot of Figure 2 shows performance for a videophone sequence known as *Mother & Daughter*, with moderate object motion and a stationary background. The lower plot of Figure 2 shows performance for a more demanding scene known as *Foreman*, with heavy object motion and an unstable hand-held moving camera. Each sequence was encoded in QCIF resolution at 10 frames per second using the framework of a well-optimized H.263 [5] video encoder. (H.263 has 16×16 prediction mode regions called macroblocks and 8×8 DCT-based DFD coding.)

A gain in performance is shown for forming a conditional replenishment (CR) coder by adding the Skip coding mode to the encoder. Further gains in performance are shown when adding various Inter coding modes to the encoder which were discussed in the previous sections:

- Inter(MV=(0,0) only): frame difference coding with only zero-valued MV displacements
- Inter(Full-pixel motion compensation): integer-pixel (full-pixel) precision motion compensation with DFD coding
- Inter(Half-pixel motion compensation): half-pixel precision motion compensation with DFD coding
- Inter & Inter+4V: half-pixel precision motion compensation with DFD coding and the addition of an "advanced prediction" mode which includes a segmentation

switch allowing a choice of either one or four MVs per 16×16 area and also includes overlapping areas of influence for MVs and extrapolation of the picture boundaries [5]. (The use of four MVs per macroblock is called the INTER+4V prediction mode.)

Except in the final case, the same H.263 baseline syntax was used throughout, with changes only in the coding method (the lower four curves are thus slightly penalized in performance by providing syntactical support for features which are never used in the encoding). In the final case, H.263 syntax was used with its Annexes D and F active [5].

However, the coding results for the two sequences differ. In the low-motion sequence, the gain achieved by using CR(a choice of SKIP or INTRA) instead of just INTRA-picture coding is the most substantial, and as more features are added, the benefits diminish. On the high-motion sequence, CR is not very useful because the whole picture is changing from frame to frame, and the addition of motion compensation using the various INTER modes provides the most significant gain, with further gain added by each increasing degree of sophistication in motion handling.

3 Optimization Techniques

In the previous section, it was demonstrated that by adding efficient coding options in the rate-distortion sense to a video codec, the overall performance increases. The optimization task is to choose, for each image region, the most efficient coded representation (segmentation, prediction modes, MVs, quantization levels, etc.) in the rate-distortion sense. This task is complicated by the fact that the various coding options show varying efficiency at different bit rates (or levels of fidelity) and with different scene content.

For example, in H.263 [5], block-based motion compensation followed by quantization of the prediction error (INTER mode) is an efficient means for coding much of the key changing content in image sequences. On the other hand, coding a particular macroblock directly (INTRA mode) may be more productive in situations when the block-based translational motion model breaks down. For relatively dormant regions of the video, simply copying a portion of the previously decoded frame into the current frame may be preferred (SKIP mode). Intuitively, by allowing multiple modes of operation, we expect improved rate-distortion performance if the modes can significantly customize the coding for different types of scene statistics, and especially if the modes can be applied judiciously to different spatial and temporal regions of an image sequence.

The modes of operation that are assigned to the image regions have differing rate-distortion characteristics, and the goal of an encoder is optimize its overall fidelity: *Minimize distortion D, subject to a constraint R_c on the number of bits used R.* This constrained problem reads as follows

$$\min D \quad \text{subject to} \quad R < R_c \; . \tag{7}$$

The optimization task in Equation (7) can be elegantly solved using Lagrangian optimization where a distortion term is weighted against a rate term [14, 15, 16, 17]. The Lagrangian formulation of the minimization problem is given by

$$\min J, \quad \text{where} \quad J = D + \lambda R, \tag{8}$$

where the Lagrangian rate-distortion functional J is minimized for a particular value of the Lagrange multiplier λ. Each solution to Equation (8) for a given value of the Lagrange multiplier λ corresponds to an optimal solution to Equation (7) for a particular value of R_c [18, 14]. More details on Lagrangian optimization are discussed in the accompanying article by A. Ortega and K. Ramchandran [19].

This technique has gained importance due to its effectiveness, conceptual simplicity, and its ability to effectively evaluate a large number of possible coding choices in an optimized fashion. If Lagrangian bit allocation and entropy-codes for signaling the coding modes are used, the number of choices available for use need not be restricted to just a few. As a result, the computation time to test all the modes may become the limiting factor on performance, rather than the capabilities of the syntax itself.

In practice, a number of interactions between coding decisions must be neglected in video coding optimization. The primary problems are the use of motion estimation and compensation and prediction mode decisions, and the common presence of cascading effects of decisions made for one region on the coding of subsequent regions in space and time. In addition, the overall bit rate must typically be controlled to match the channel capacity – which further complicates matters. All three quantities, D, λ, and R tend to be subject to approximations and compromises in designing video coding systems.

3.1 Bit Rate Control

The overall bit rate of a video coder is determined by its prediction mode decisions, MV choices, and DFD coding fidelity. The last of these three is typically the most important for bit rate control, and the residual fidelity is typically controlled by choosing a step size scaling to be used for inverse quantization reconstruction of the transformed difference signal [20]. A larger step size results in a lower bit rate and a larger amount of distortion. Thus the choice of step size is closely related to the choice of the relative emphasis to be placed on rate and distortion, i.e., the choice of λ. (The choice of the quantizer step size scaling must be communicated to the decoder, but λ is an encoder-only issue and is not needed by the decoder.) As a last resort, the coding of entire pictures can be skipped by the encoder as a bit rate control mechanism (resulting in a less fluid rendition of motion).

In some cases the bit rate must be controlled to maintain a constant local-average bit rate over time, but in other cases it may be allowed to vary much more widely (such as by allowing the amount of scene content activity to govern the bit rate). Whatever the constraints imposed on the bit rate of the system, control over λ in a well-optimized encoder can provide an excellent means of meeting those constraints. In a later section we will show how control over λ can be tightly linked to the more conventional practice of control over the inverse quantization step size.

A feedback control of the buffer state of video codecs was proposed by Choi and Park in [21], where the control is applied to the Lagrange multiplier λ. Trellis-based buffer control has been presented by Ortega, Ramchandran, and Vetterli in [22], where fast approximations are achieved using the Lagrangian formulation. A low-delay rate control method for H.263 was provided in [23]. There are many approaches to rate control; However, the use of the Lagrange multiplier method of optimization within these rate control schemes can often help to avoid losses in coding performance that might otherwise result from their use.

3.2 Motion Estimation

Ideally, decisions should be controlled by their ultimate effect on the resulting pictures, however this ideal may not be attainable in an encoder implementation. For example, in considering each possible MV to send for a picture area, an encoder should perform an optimized coding of the residual error and measure the resulting bit usage and distortion. Only by doing this can it really choose the best possible MV value to send (even if neglecting the effect of that choice on later choices spatially and later pictures temporally). However, there are typically thousands of possible MV values to choose from, and coding just one residual difference signal typically requires a significant fraction of the total computational power of a practical encoder.

A simpler method of performing motion estimation is to simply search for a MV which minimizes the prediction error prior to residual coding, perhaps giving some special preference to the zero-valued MV and to the MV value which requires the fewest bits to represent as a result of MV prediction in the decoder. These biases prevent spurious large MV values (which require a large number of bits to represent but may provide only little prediction benefit).

Further simplification is needed in real-time implementations. A straightforward minimum-squared-error "full-search" motion estimation which tests all possible integer values of a MV within a $\pm L$ range (video coding syntax typically supports $L = 16$ or $L = 32$, and one optional mode of H.263 supports an unlimited range) would require approximately $3(2L+1)^2$ operations per pixel (two adds and one multiply per tested MV value). Adding half-pixel MVs to the search multiplies the number of MV values to test by a factor of four, and adds the requirement of an interpolation operation for generating the half-pixel sampling grid locations in the prior picture. Such levels of complexity are beyond the capabilities of many of today's video coder implementations – and if this much computational power was available to an implementation, devoting it all to this type of search might not be the best way to gain performance. Motion estimation complexity is often reduced in implementations by the use of iterative refinement techniques. While we do not specifically address reduced-complexity motion estimation herein, rate-distortion optimization within the context of a reduced-complexity search can also often provide a performance benefit.

We can view MCP formation as a source coding problem with a fidelity criterion, closely related to vector quantization. For the number of bits required to transmit the MVs, MCP provides a version of the video signal with a certain fidelity. The rate-distortion trade-off can be controlled by various means. One approach is to treat MCP as *entropy-constrained vector quantization* (ECVQ) [15, 16]. Here, each image block to be encoded is quantized using its own codebook that consists of a neighborhood of image blocks of the same size in the previously decoded frames (as determined by the motion estimation search range). A codebook entry is addressed by the translational MVs which are entropy-coded. The criterion for the block motion estimation is the minimization of a Lagrangian cost function wherein the distortion, represented as the prediction error in SSD or SAD, is weighted against the number of bits associated with the translational MVs using a Lagrange multiplier.

An alternative interpretation is to view the motion search as an estimation problem: the estimation of a motion displacement field for the image. The problem of motion estimation becomes increasingly ill-conditioned as we increase the motion estimation search range and reduce the block size. The ill-conditioning results in a lack of consistency in the estimated MVs, resulting in a loss of accuracy in estimating true motion. The Lagrangian formulation can regularize the displacement field estimate. Hence, the Lagrangian formulation yields a solution to the problem not only when viewing motion estimation as a source coding technique, but also when viewing it an ill-conditioned displacement field estimation problem.

Block motion estimation can therefore be viewed as the minimization of the Lagrangian cost function

$$J_{\mathrm{MOTION}} = D_{\mathrm{DFD}} + \lambda_{\mathrm{MOTION}} R_{\mathrm{MOTION}}, \tag{9}$$

in which the distortion D_{DFD}, representing the prediction error measured as SSDor SAD, is weighted against the number of bits R_{MOTION} associated with the MVs using a Lagrange multiplier $\lambda_{\mathrm{MOTION}}$. The Lagrange multiplier imposes the rate constraint as in ECVQ, and its value directly controls the rate-distortion trade-off, meaning that small values of $\lambda_{\mathrm{MOTION}}$ correspond to high fidelities and bit rates and large values of $\lambda_{\mathrm{MOTION}}$ correspond to lower fidelities and bit rates. Sullivan and Baker proposed such a rate-distortion optimized motion estimation scheme for fixed or variable block sizes in [16], and more work on the subject has appeared in [17, 24, 25, 26, 27].

3.2.1 Variable Block Sizes

The impact of the block size on MCP fidelity and bit rate are illustrated in Figure 3 for the video sequences *Mother & Daughter* (top) and *Foreman* (bottom). For the data in this figure, the motion estimation and compensation were performed using the sequence of original video frames, with temporal sub-sampling by a factor of 3. The motion estimation was performed by minimizing J_{MOTION} in Equation (9). In the first part of the motion estimation procedure, an integer-pixel accuracy displacement vector was found within a search range of $[-16..16] \times [-16..16]$ pixels relative to the location of the block to be searched. Then, given this integer-pixel accurate displacement vector, its surrounding half-pixel positions were checked for improvements when evaluating Equation (9). This second stage of this process is commonly called half-pixel refinement.

For the curves in Figure 3, the same H.263 (with Annexes D and F) motion compensation syntax was used throughout, with changes only in the encoding method. These changes are:

- Case 1: INTER-coding using only luminance blocks of size 16×16 samples (SKIP and

INTER modes)

- Case 2: INTER-coding using only luminance blocks of size 8 × 8 samples (SKIP and INTER+4V modes)
- Case 3: Combining cases 1 and 2 using a rate-constrained encoding strategy, which adapts the frequency of using the various block sizes using Lagrange multiplier optimization [16].

Comparing cases 1 and 2, the use of 16 × 16 blocks is more beneficial at low rates, while 8 × 8 blocks provide more coding gain at high bit rates. However, case 3 provides superior coding performance for all bit rates.

The ultimate impact of the block size on the final objective coding fidelity is shown in Figure 4. In this experiment, the residual coding stage and the INTRA coding mode were added to the scheme, producing a complete encoder. The tendencies observed for the case of motion compensation only (see Figure 3) are also true here. Allowing a large range of coding fidelities for MCP provides superior performance over the entire range of bit rates. However, every extra bit spent for motion compensation must be justified against other coding options like residual coding [17].

3.2.2 Other Methods for Improving Motion-Compensated Prediction

Besides block size variation to improve the MCP, various other methods have been proposed. Examples of these schemes include

1. Multi-hypothesis MCP,
2. Long-term memory MCP,
3. Complex motion models,

The idea of the first item, multi-hypothesis MCP, is that various signals are superimposed to compute the MCP signal. The multi-hypothesis motion-compensated predictor for a pixel location $\boldsymbol{s} \in \boldsymbol{\mathcal{A}}_{i,n}$ is defined as

$$\widehat{\boldsymbol{I}}_n(\boldsymbol{s}) = \sum_{p=1}^{P} h_p(\boldsymbol{s}) \cdot \widetilde{\boldsymbol{I}}_{n-\Delta n}(\boldsymbol{s} - \boldsymbol{v}_{i,n,p}) \ , \ \ \boldsymbol{s} \in \boldsymbol{\mathcal{A}}_{i,n} \ . \tag{10}$$

with $\widehat{\boldsymbol{I}}_n(\boldsymbol{s})$ being a predicted pixel value and $\widetilde{\boldsymbol{I}}_{n-\Delta n}(\boldsymbol{s} - \boldsymbol{v}_{i,n,p})$ being a motion-compensated pixel from a decoded frame Δn time instants in the past (normally $\Delta n = 1$). This scheme is a generalization of Equation (1) and it includes concepts like sub-pixel accurate MCP [28, 29], B-frames [30], spatial filtering [3] and overlapped block motion compensation (OBMC) [9, 10, 11]. Using the linear filtering approach of Equation (10), the accuracy of motion compensation can be significantly improved. An estimation-theoretic analysis of multi-hypothesis MCP was presented in [11]. A rate-distortion efficiency analysis of multi-hypothesis motion compensation including OBMC and B-frames was presented in [31].

The second item, long-term memory MCP as proposed in [32], refers to extending the spatial displacement vector utilized in block-based hybrid video coding by a variable time delay, permitting the use of more frames than the last prior decoded one for MCP. The long-term memory covers several seconds of decoded frames at encoder and decoder. Experiments when employing 50 frames for motion compensation using the sequences *Foreman* and *Mother & Daughter* show that long-term memory MCP yields about 2 dB and 1 dB PSNR improvements in prediction error against the 1 frame case, respectively. However, in the long-term memory case, the MV bit rate shows an increase of 30% compared to the one frame case [32]. Embedded in a complete video coder, the approach still yields significant coding gains expressed in bit rate savings of 23% for the sequence *Foreman* and 17% for the sequence *Mother & Daughter* due to the impact of long-term memory MCP when comparing it to the rate-distortion optimized H.263 coder which is outlined in this paper [32].

Complex motion models, the third item, have been proposed by a great number of researchers for improving motion compensation performance. The main effect is increased accuracy using a higher order approximation of the displacement vector field (e.g., using polynomial motion models) than the accuracy achievable with translational motion models which relate to piecewise constant approximation. In [33] and [34], a complete video codec is presented, where image segments are motion-compensated using bilinear (12 parameter) motion models. The image segments partition a video frame down to a granularity of 8×8 blocks. Bit rate savings of more than 25% were reported for the sequence *Foreman* [34].

3.3 Intra/Inter/Skip Mode Decision

Hybrid video coding consists of the motion estimation and the residual coding stages, and an interface between them consisting of prediction mode decision. The task for the residual coding is to represent signal parts that are not sufficiently approximated by the earlier stages. From the view-point of bit-allocation strategies, the various prediction modes relate to various bit rate partitions. Considering the various H.263 modes: Intra, Skip, Inter, and INTER+4V, Table 1 gives typical values for the bit rate partition of motion and DFD texture coding for typical sequences. The various modes in Table 1 relate to quite different overall bit rates. Since the choice of mode is adapted to the scene content it is transmitted as side information.

If we assume for the moment that the bit rate and distortion of the residual coding stage is controlled by the selection of a quantizer step size Q, then rate-distortion optimized mode decision refers to the minimization of the following Lagrangian functional

$$J(\boldsymbol{\mathcal{A}}, M, Q) = D_{\mathrm{REC}}(\boldsymbol{\mathcal{A}}, M, Q) + \lambda_{\mathrm{MODE}} R_{\mathrm{REC}}(\boldsymbol{\mathcal{A}}, M, Q), \tag{11}$$

where, for instance, $M \in \{\textsc{Intra}, \textsc{Skip}, \textsc{Inter}, \textsc{Inter}{+}4\mathrm{V}\}$ indicates a mode chosen for a particular macroblock, Q is the selected quantizer step size, $D_{\mathrm{REC}}(\boldsymbol{\mathcal{A}}, M, Q)$ is the SSD between the original macroblock $\boldsymbol{\mathcal{A}}$ and its reconstruction, and $R_{\mathrm{REC}}(\boldsymbol{\mathcal{A}}, M, Q)$ is the number of bits associated with choosing M and Q.

A simple algorithm for rate-constrained mode decision minimizes Equation (11) given all mode decisions of past macroblocks. This procedure partially neglects dependencies between macroblocks like prediction of MV values from those of neighboring blocks and OBMC. In [35] and in [36], Wiegand et al. proposed the exploitation of dependencies between macroblocks using dynamic programming methods. Later work on the subject which also included the option to change the quantizer value on a macroblock to macroblock basis appeared by Schuster and Katasggelos [37].

3.4 Quantization

After DCT transformation, the residual signal must be quantized to form the final estimate. Ideally, the choice of quantizer step size Q should be optimized in a rate-distortion sense. Given a quantizer step size Q, the quantization of the residual signal (the mapping of the transformed samples to quantization index values) should also be rate-distortion optimized. The choice of the quantizer output level sent for a given input value should balance the needs of rate and distortion. A simple way to do this is to move the decision thresholds of the quantizer somewhat toward the lower bit rate index [38]. This is the method used in the ITU-T test model [20]. Alternatively, a $D + \lambda R$ decision can be made explicitly to choose the quantization index. However, in modern video coders such as H.263 the bit rate needed to represent a given quantization index depends not only on the index chosen for a particular sample, but on the values of neighboring quantized indices as well (due to the structure of the coefficient index entropy coding method used). The best performance can be obtained by accounting properly for these interactions, as can be achieved by using a trellis-based quantization technique. Such a quantization scheme was proposed by Ortega and Ramchandran [39] and a version which handles the more complex structure of the entropy coding of H.263 has recently appeared [40, 41]. Trellis-based quantization can provide approximately a 3% reduction in the bit rate needed for a given level of fidelity (when applied to H.263-based DCT coding).

3.5 Choosing λ and the Quantization Step Size Q

The algorithm for the rate-constrained mode decision can be modified in order to incorporate macroblock quantization step size changes. For that, the set of macroblock modes to choose from can be extended by also including the prediction mode type INTER+Q for each macroblock, which permits changing Q by a small amount when sending an INTER macroblock. More precisely, for each macroblock a mode M can be chosen from the set

$$\begin{aligned} M \in \quad & \{\textsc{Intra}, \textsc{Skip}, \textsc{Inter}, \textsc{Inter}{+}4\text{V}, \ldots \\ & \textsc{Inter}{+}\text{Q}(-2), \textsc{Inter}{+}\text{Q}(-1), \textsc{Inter}{+}\text{Q}(+1), \textsc{Inter}{+}\text{Q}(+2)\}, \end{aligned} \tag{12}$$

where, for example, INTER+Q(-1) stands for the INTER mode being coded with quantizer value reduced by one relative to the previous macroblock. Hence, the macroblock Q selected by the minimization routine becomes dependent on λ_{MODE}. Otherwise the algorithm for running the rate-distortion optimized coder remains unchanged.

Figure 5 show the relative occurrence of macroblock QUANT values (in H.263, QUANT is $Q/2$) for several Lagrange parameter settings. The Lagrange parameter λ_{MODE} is varied over seven values: 4, 25, 100, 250, 400, 730, 1000, producing seven normalized histograms that are depicted in the plots in Figure 5. In Figure 5, the macroblock QUANT values are gathered while coding 100 frames of the video sequences *Foreman*, *Mobile & Calendar*, *Mother & Daughter*, and *News*.

Note that for small macroblock quantizer values, the initial quantizer value setting does not greatly affect the outcome of the experiment. However, for large values, i.e., low bit rates, some impact of the initial quantizer value can be observed, since changing the quantizer value on a macroblock basis is combined with a bit rate penalty in H.263. Nevertheless, the same tendencies can be observed but require a longer duration of the video sequences than the ones used for the results in Figure 5.

Figure 6 shows the obtained average macroblock $\textsc{Quant}$ gathered when coding the complete sequences *Foreman*, *Mobile & Calendar*, *Mother & Daughter*, and *News*. The red curve relates to the function

$$\lambda_{\text{MODE}} = 0.85 \cdot (\textsc{Quant})^2 \ , \tag{13}$$

which is an approximation of the functional relationship between the macroblock $\textsc{Quant}$ and the Lagrange parameter λ_{MODE} up to $\textsc{Quant}$ values of 25, and H.263 allows only a choice of $\textsc{Quant} \in \{1, 2, \ldots, 31\}$. Particularly remarkable is the strong dependency between λ_{MODE} and $\textsc{Quant}$, even for sequences with widely varying content. Note, however, that for a given value of λ_{MODE}, the chosen $\textsc{Quant}$ tends to be higher for sequences which require higher amounts of bits (*Mobile & Calendar*) in comparison to sequences requiring smaller amounts of bits for coding at that particular λ_{MODE} (*Mother & Daughter*) – but these differences are rather small.

As a further justification of our simple approximation of the relationship between λ_{MODE} and Q, let us assume a typical quantization curve high-rate approximation [42] as follows

$$R(D) = a \ln \left(\frac{\sigma^2}{D} \right) . \tag{14}$$

Taking the derivative of $R(D)$ with respect to D and setting its value to $-1/\lambda_{\text{MODE}}$ yields

$$\frac{dR(D)}{dD} = -\frac{a}{D} \overset{\triangle}{=} -\frac{1}{\lambda_{\text{MODE}}}. \tag{15}$$

Noting that at high rates $D \cong (2\textsc{Quant})^2/12$, we get

$$\lambda_{\text{MODE}} \cong c \cdot (\textsc{Quant})^2 \ , \tag{16}$$

where $c = 4/(12a)$. Although our assumptions may not be completely realistic, the derivation reveals at least the qualitative insight that it may be reasonable for the value of the Lagrange parameter λ_{MODE} to be proportional to the square of the quantization parameter. As shown above, 0.85 appears to be a reasonable value for use as the constant c.

A low complexity rule is needed for motion estimation, so we alter the rule slightly for that part of the encoding optimization. In order to reduce the computational requirements of motion estimation, we prefer to use the SAD measure rather than the SSD measure in that stage of encoding. Experimentally, we have found that an effective such method is to measure distortion using SAD and to simply adjust λ for the lack of the squaring operation in the error computation, as given by

$$\lambda_{\text{MOTION}} = \sqrt{\lambda_{\text{MODE}}} \ . \tag{17}$$

This strong dependency that we have thus derived between QUANT, λ_{MODE}, and λ_{MOTION} offers a simple treatment of each of these quantities as a dependent variable of another. For example, the rate control method may adjust the macroblock QUANT occasionally so as to control the average bit rate of a video sequence, while treating λ_{MODE} and λ_{MOTION} as dependent variables using Equations (13) and (17). In the experiments reported herein, we therefore used the approximation refmotionlambda with the SAD error measure for motion estimationand the approximation refmodelambda with the SSD error measure for mode decisions.

4 Comparison to Other Encoding Strategies

The ITU-T video coding experts group (ITU-T Q.15/SG16) maintains an internal document describing examples of encoding strategies, called its test model [43, 20]. The mode decision and motion estimation optimization strategies described above along with the method of choosing λ_{MODE} based on quantizer step size as shown above were recently proposed by the second author and others for inclusion into this test model [44, 45]. The group, which is chaired by the first author, had previously been using a less-optimized encoding approach for its internal evaluations [43], but accepted these methods in the creation of a more recent model [20]. The test model documents, the other referenced Q.15 documents, and other information relating to ITU-T video coding experts group work can be found on an ftp site maintained by the group (`ftp://standard.pictel.com/video-site`). Reference software for the test model is available by ftp from the University of British Columbia (`ftp://dspftp.ee.ubc.ca/pub/tmn`, with further information at `http://www.ece.ubc.ca/spmg/research/motion/h263plus`).

The less sophisticated TMN-9 mode decision method is based on thresholds. It compared the sum of absolute differences of the 16×16 macroblock (W) with respect to its mean value to the minimum prediction SAD obtained by an integer-pixel motion search in order to make its decision between INTRA and INTER modes as follows

$$W \overset{?}{<} \text{minSAD}(\text{full-pixel}, 16 \times 16) - 500. \tag{18}$$

When this inequality was satisfied, the INTRA mode would be chosen for that particular macroblock. The minSAD value above corresponds to the minimum SAD value after integer-pixel motion compensation using a 16×16 motion compensation block size, where the SAD value of the $(0,0)$ MV is reduced by 100 to bias the decision toward choosing the SKIP mode. If the INTER mode is chosen, i.e., the inequality above is not satisfied, the chosen integer-pixel MV was half-pixel refined. The MVs for INTER+4V blocks were found by half-pixel refining the integer pixel MV of the 16×16 blocks. Finally, the INTER+4V mode was chosen if

$$\sum_{i=0}^{3} \mathrm{minSAD}_i(\text{half-pixel}, 8 \times 8) < \mathrm{minSAD}(\text{half-pixel}, 16 \times 16) - 200. \tag{19}$$

was satisfied, where $\mathrm{minSAD}_i(\text{half-pixel}, 8 \times 8)$ is the minimum SAD value of the i'th of the four 8×8 blocks. The SKIP mode was chosen in TMN-9 only if the INTER mode was chosen as better than the INTRA mode and the MV components and all of the quantized transform coefficients were zero.

In the TMN-10 rate-distortion optimized strategy, the motion search uses rate-constrained motion estimation for first finding the best integer-pixel MV in the search range of ±15 pixels. Then, the best inter-pixel MV is half-pixel refined again by minimizing the Lagrangian cost functional for motion estimation given in Equation (9). This procedure is executed for 16×16 and 8×8 blocks. The mode decision of TMN-10 is then conducted using the rate-distortion optimized method described in this article.

The role of the encoding strategy is demonstrated in Figure 7 for the video sequences *Mother & Daughter* and *Foreman*. The same syntax (H.263 using Annexes D and F) was used throughout, with changes only in the mode decision and motion estimation coding methods. These changes are:

- Case 1: TMN-9 mode decision and TMN-9 motion estimation
- Case 2: TMN-10 mode decision and TMN-9 motion estimation
- Case 3: TMN-10 mode decision and TMN-10 motion estimation

Case 2 has been included to demonstrate the impact of rate-constrained mode decision and motion estimation separately. Comparing the three cases, we find that the usage of the full motion estimation search range of ±15 pixels for the 8×8 block displacement vectors in INTER+4V mode provides most of the gain for the TMN-10 encoding strategy. The INTER+4V prediction mode is very seldom used in TMN-9, indicating that the TMN-9 motion estimation and mode decision rules basically fail to make effective use of this mode. In the highly active *Foreman* sequence, TMN-10 (Case 3) uses this mode for about 15% of macroblocks, whereas TMN-9 (Case 1) uses it for only about 2%.

The TMN-9 motion estimation strategy only permits the use of half-pixel positions for the 8 × 8 block displacement vectors that surround the previously found 16 × 16 block displacement vector, which is searched in the ±15 range. We have observed that using the full search range for the 8 × 8 block displacement vectors leads to improved coding performance for the rate-constrained motion estimation, whereas for the TMN-9 motion estimation, using the full search for this small block size would actually harm the TMN-9 results, since no rate constraint was employed in its search. Only adding a rate constraint to the motion estimation can allow the INTER+4V mode be perform with its full potential.

Figure 8 shows that the TMN-10 coder uses about twice as many bits for motion than the other two coders in order to obtain a better prediction so it can use less difference coding and still obtain an improvement in the overall performance. This is partly because of more frequent use of the INTER+4V mode and partly because of the larger motion estimation search range considered for the 8 × 8 blocks when the INTER+4V mode is chosen.

In the TMN-10 strategy, the bit rate allocated to the motion part of the information increases as the overall bit rate increases, which makes intuitive sense. The TMN-9 motion estimation shows completely different and sometimes counterintuitive behavior. For the sequence *Foreman*, the MV bit rate actually decreases as overall bit rate increases. This results from the facts that the TMN-9 motion estimation does not employ a rate constraint and that motion estimation is performed using the reconstructed frames (for TMN-9 as well as for TMN-10). As bit rate decreases these reconstructed frames get noisier and since the regularization by the rate constraint is missing for the TMN-9 motion estimation, the estimates for the motion data get noisier and require a higher bit rate.

Rate-constrained mode decision, as employed in TMN-10, provides rather minor gains, but also introduces a reasonably small computational overhead. The overall performance gain of the improved mode decision and motion estimation methods is typically around 10% in bit rate, or 0.5 dB in PSNR.

5 Conclusions

We have described the structure of typical video coders, and showed that their design and operation requires a keen understanding and analysis of the trade-offs between bit rate and distortion. The single powerful principle of $D + \lambda R$ Lagrange multiplier optimization [14] has emerged as the weapon of choice in the optimization of such systems, and can provide significant benefits if judiciously applied.

6 Acknowledgements

The authors wish to thank Klaus Stuhlmüller, Niko Färber, Bernd Girod, Barry Andrews, and the PictureTel research team for their support and useful discussions. They also wish to thank the guest editors Antonio Ortega and Kannan Ramchandran, as well as John Villasenor and his UCLA research team, Faouzi Kossentini and his UBC research team, and Jonathan Su for their valuable comments on early manuscripts for this article.

References

[1] ITU-T (formerly CCITT) and ISO/IEC JTC1, "Digital Compression and Coding of Continuous-Tone Still Images", ISO/IEC 10918-1 — ITU-T Recommendation T.81 (JPEG), Sept. 1992.

[2] ITU-T (formerly CCITT), "Codec for Videoconferencing Using Primary Digital Group Transmission", ITU-T Recommendation H.120; version 1, 1984; version 2, 1988.

[3] ITU-T (formerly CCITT), "Video Codec for Audiovisual Services at $p \times 64$ kbit/s", ITU-T Recommendation H.261; version 1, Nov., 1990; version 2, Mar., 1993.

[4] ISO/IEC JTC1, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mbit/s — Part 2: Video", ISO/IEC 11172-2 (MPEG-1), Mar. 1993.

[5] ITU-T (formerly CCITT), "Video Coding for Low Bitrate Communication", ITU-T Recommendation H.263; version 1, Nov., 1995; version 2, Jan., 1998.

[6] B. Girod, E. Steinbach, and N. Färber, "Performance of the H.263 Video Compression Standard", *Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology*, 1997.

[7] ITU-T (formerly CCITT) and ISO/IEC JTC1, "Generic Coding of Moving Pictures and Associated Audio Information — Part 2: Video", ITU-T Recommendation H.262 — ISO/IEC 13818-2 (MPEG-2), Nov. 1994.

[8] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "Optimal Shape Coding Techniques", *IEEE Signal Processing Magazine*, Nov. 1998, (this issue).

[9] H. Watanabe and S. Singhal, "Windowed Motion Compensation", in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, 1991, vol. 1605, pp. 582–589.

[10] S. Nogaki and M. Ohta, "An Overlapped Block Motion Compensation for High Quality Motion Picture Coding", in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1992, vol. 1, pp. 184–187.

[11] M. T. Orchard and G. J. Sullivan, "Overlapped Block Motion Compensation: An Estimation-Theoretic Approach", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 693–699, Sept. 1994.

[12] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Coders", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533–545, Sept. 1994.

[13] J. Lee and B. W. Dickinson, "Joint Optimization of Frame Type Selection and Bit Allocation for MPEG Video Coders", in *Proceedings of the IEEE International Conference on Image Processing*, Austin, USA, Nov. 1994, vol. 2, pp. 962–966.

[14] Y. Shoham and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

[15] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-Constrained Vector Quantization", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 1, pp. 31–42, Jan. 1989.

[16] G. J. Sullivan and R. L. Baker, "Rate-Distortion Optimized Motion Compensation for Video Compression Using Fixed or Variable Size Blocks", in *GLOBECOM'91*, 1991, pp. 85–90.

[17] B. Girod, "Rate-Constrained Motion Estimation", in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Chicago, USA, Sept. 1994, pp. 1026–1034, (invited paper).

[18] H. Everett III, "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources", *Operations Research*, vol. 11, pp. 399–417, 1963.

[19] A. Ortega and K. Ramchandran, "Rate-Distortion Methods for Image and Video Compression: An Overview", *IEEE Signal Processing Magazine*, Nov. 1998, (this issue).

[20] ITU-T, SG15/WP15/1, Q15-D-65, "Video Codec Test Model Number 10 (TMN-10)", Download vis anonymous ftp to standard.pictel.com, Apr. 1998.

[21] J. Choi and D. Park, "A Stable Feeedback Control of the Buffer State Using the Controlled Lagrange Multiplier Method", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 546–558, Sept. 1994.

[22] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal Trellis-Based Buffered Compression and Fast Approximations", *IEEE Transactions on Image Processing*, vol. 3, no. 1, pp. 26–40, Jan. 1994.

[23] ITU-T, SG15/WP15/1, Q15-A-20, J. Ribas-Corbera and S. Lei, "Rate Control for Low-Delay Video Communications", Download vis anonymous ftp to standard.pictel.com, June 1997.

[24] J. Lee, "Optimal Quadtree for Variable Block Size Motion Estimation", in *Proceedings of the IEEE International Conference on Image Processing*, Washington, D.C., USA, Oct. 1995, vol. II, pp. 480–483.

[25] W. C. Chung, F. Kossentini, and M. J. T. Smith, "An Efficient Motion Estimation Technique Based on a Rate-Distortion Criterion", in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Atlanta, USA, May 1996.

[26] F. Kossentini, Y.-W. Lee, M. J. T. Smith, and R. Ward, "Predictive RD Optimized Motion Estimation for Very Low Bit Rate Video Coding", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp. 1752–1763, Dec. 1997.

[27] M. C. Chen and A. N. Willson, "Rate-Distortion Optimal Motion Estimation Algorithms for Motion-Compensated Transform Video Coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 2, pp. 14–158, Apr. 1998.

[28] B. Girod, "The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video Sequences", *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140–1154, Aug. 1987.

[29] B. Girod, "Motion-Compensating Prediction with Fractional-Pel Accuracy", *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[30] H. G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding", *Proceedings of the IEEE*, vol. 73, no. 9, pp. 523–548, Apr. 1985.

[31] B. Girod, "Efficiency Analysis of Multi-Hypothesis Motion-Compensated Prediction for Video Coding", *IEEE Transactions on Image Processing*, 1997, Submitted for publication.

[32] T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction", *IEEE Transactions on Circuits and Systems for Video Technology*, Sept. 1998.

[33] ISO/IEC JTC1/SC29/WG11 MPEG96/M0904, "Nokia research center: Proposal for efficient coding", Submitted to Video Subgroup, July 1996.

[34] M. Karczewicz, J. Niewęgłowski, and P. Haavisto, "Video Coding Using Motion Compensation with Polynomial Motion Vector Fields", *Signal Processing: Image Communication*, vol. 10, pp. 63–91, 1997.

[35] T. Wiegand, M. Lightstone, T.G. Campbell, and S.K. Mitra, "Efficient Mode Selection for Block-Based Motion Compensated Video Coding", in *Proceedings of the IEEE International Conference on Image Processing*, Washington, D.C., USA, Oct. 1995.

[36] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182–190, Apr. 1996.

[37] G. M. Schuster and A. K. Katsaggelos, "Fast and Efficient Mode and Quantizer Selection in the Rate Distortion Sense for H.263", in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Orlando, USA, Mar. 1996, pp. 784–795.

[38] G. J. Sullivan, "Efficient Scalar Quantization of Exponential and Laplacian Random Variables", *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1365–1374, Sept. 1996.

[39] A. Ortega and K. Ramchandran, "Forward-Adaptive Quantization with Optimal Overhead Cost for Image and Video Coding with Applications to MPEG Video Coders", in *Proceedings of the SPIE, Digital Video Compression: Algorithms and Technologies*, San Jose, USA, Feb. 1995.

[40] ITU-T, SG15/WP15/1, Q15-D-40, J. Wen, M. Luttrell, and J. Villasenor, "Simulation Results on Trellis-Based Adaptive Quantization", Download vis anonymous ftp to standard.pictel.com, Apr. 1998.

[41] J. Wen, M. Luttrell, and J.Villasenor, "Trellis-Based R-D Optimal Quantization in H.263+", *IEEE Transactions on Image Processing*, 1998, submitted for publication.

[42] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, USA, 1984.

[43] ITU-T, SG15/WP15/1, Q15-D-65, "Video Codec Test Model Number 9 (TMN-9)", Download vis anonymous ftp to standard.pictel.com, Dec. 1997.

[44] ITU-T, SG15/WP15/1, Q15-D-13, T. Wiegand and B. D. Andrews, "An Improved H.263 Coder Using Rate-Distortion Optimization", Download vis anonymous ftp to standard.pictel.com, Apr. 1998.

[45] ITU-T, SG15/WP15/1, Q15-D-49, M. Gallant, G. Cote, and F. Kossentini, "Description of and Results for Rate-Distortion-Based Coder", Download vis anonymous ftp to standard.pictel.com, Apr. 1998.

Editorial Note: The figures below are provided in color PostScript form and may not be fully appreciated on a black & white laser print-out. Use a color printer or a color PostScript viewer for proper viewing.

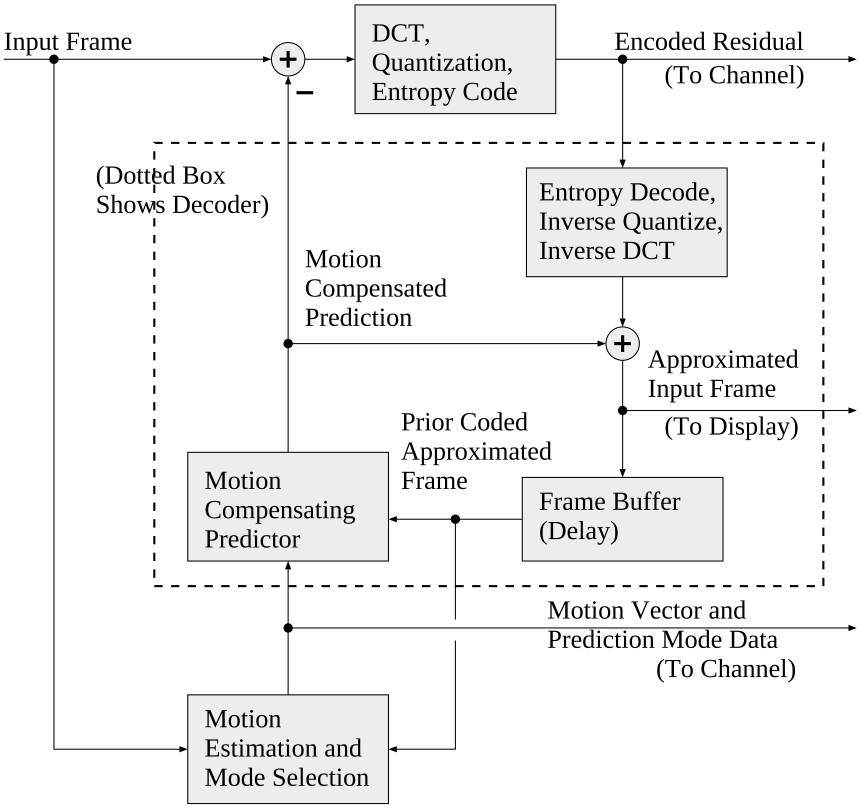

Figure 1: Typical Motion Compensated DCT Video Coder

Mode	Motion Coding Bit Rate [%]	Texture Coding Bit Rate [%]
INTRA	0	100
SKIP	100	0
INTER	30 ± 15	70 ∓ 15
INTER+4V	50 ± 20	50 ∓ 20

Table 1: Bit rate partition of the various H.263 modes.

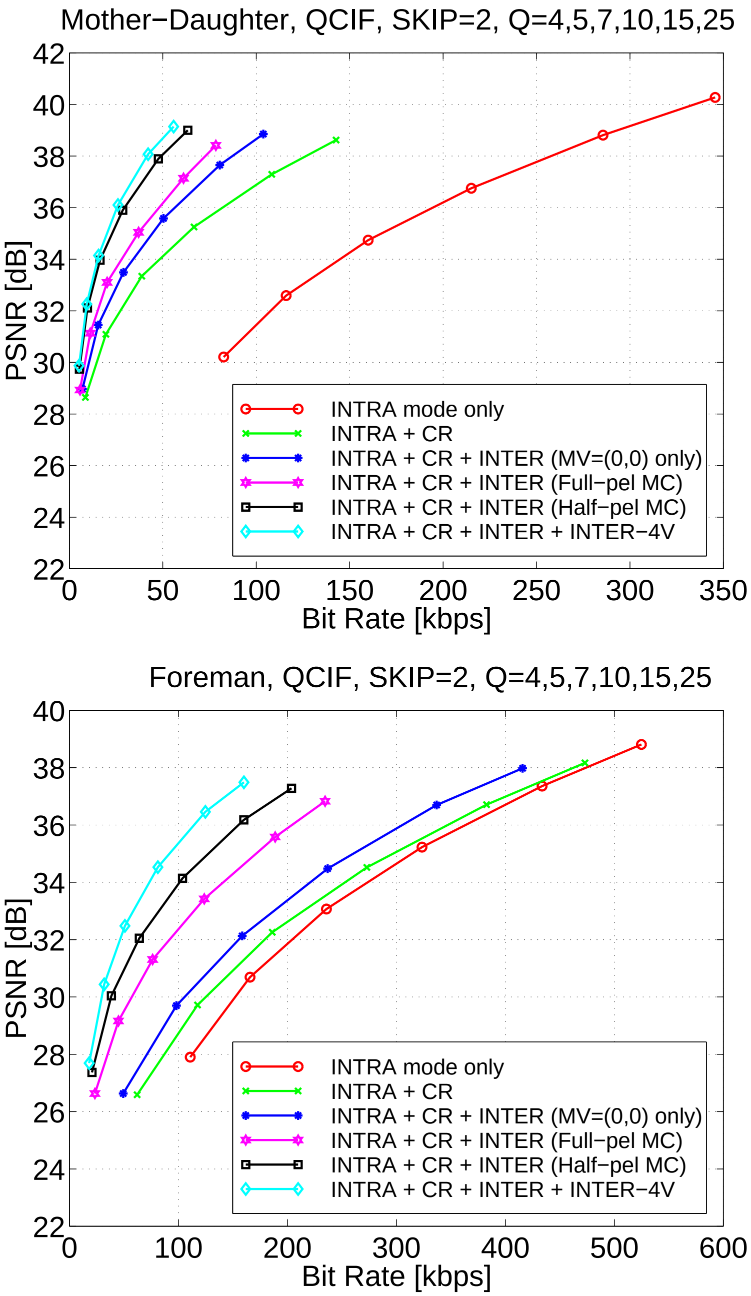

Figure 2: Coding performance for the sequence *Mother & Daughter* (top) and *Foreman* (bottom).

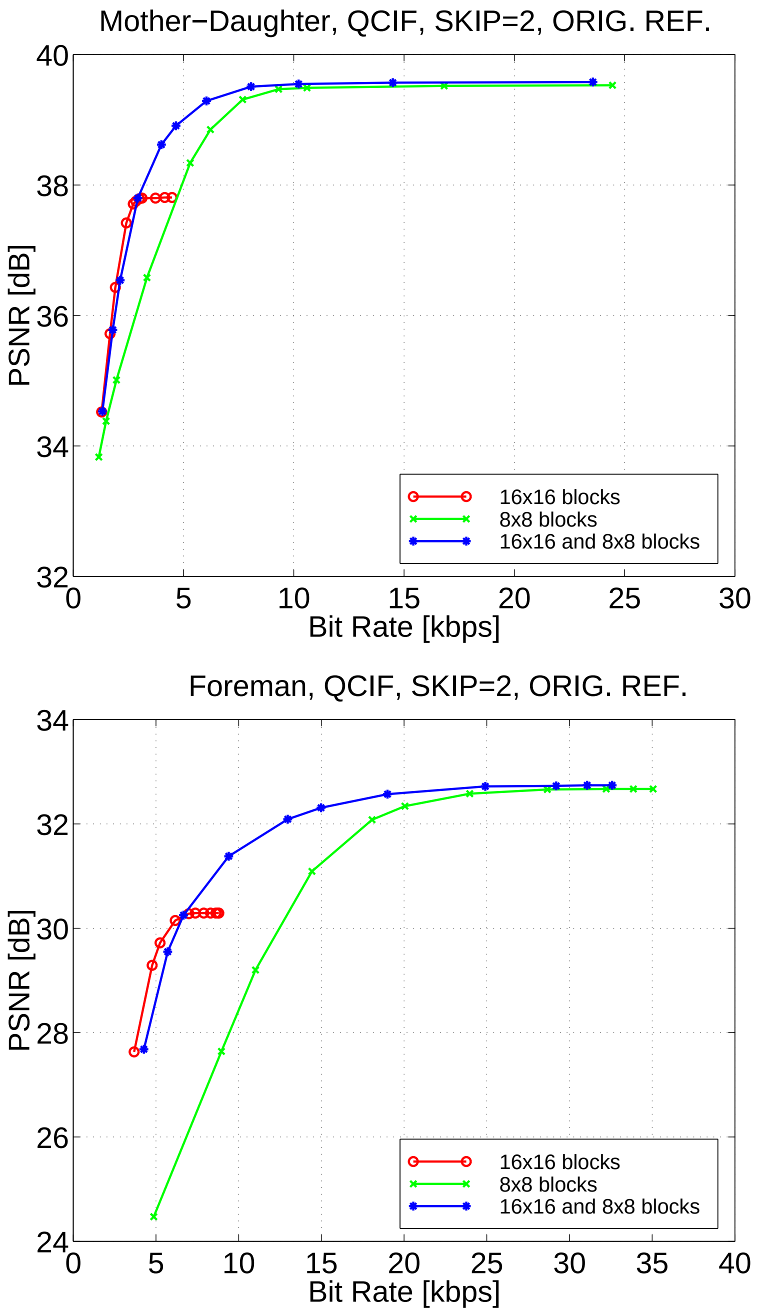

Figure 3: Prediction gain vs. MV bit rate for the sequence *Mother & Daughter* (top) and *Foreman* (bottom) when employing H.263 MV median prediction and original frames as reference frames.

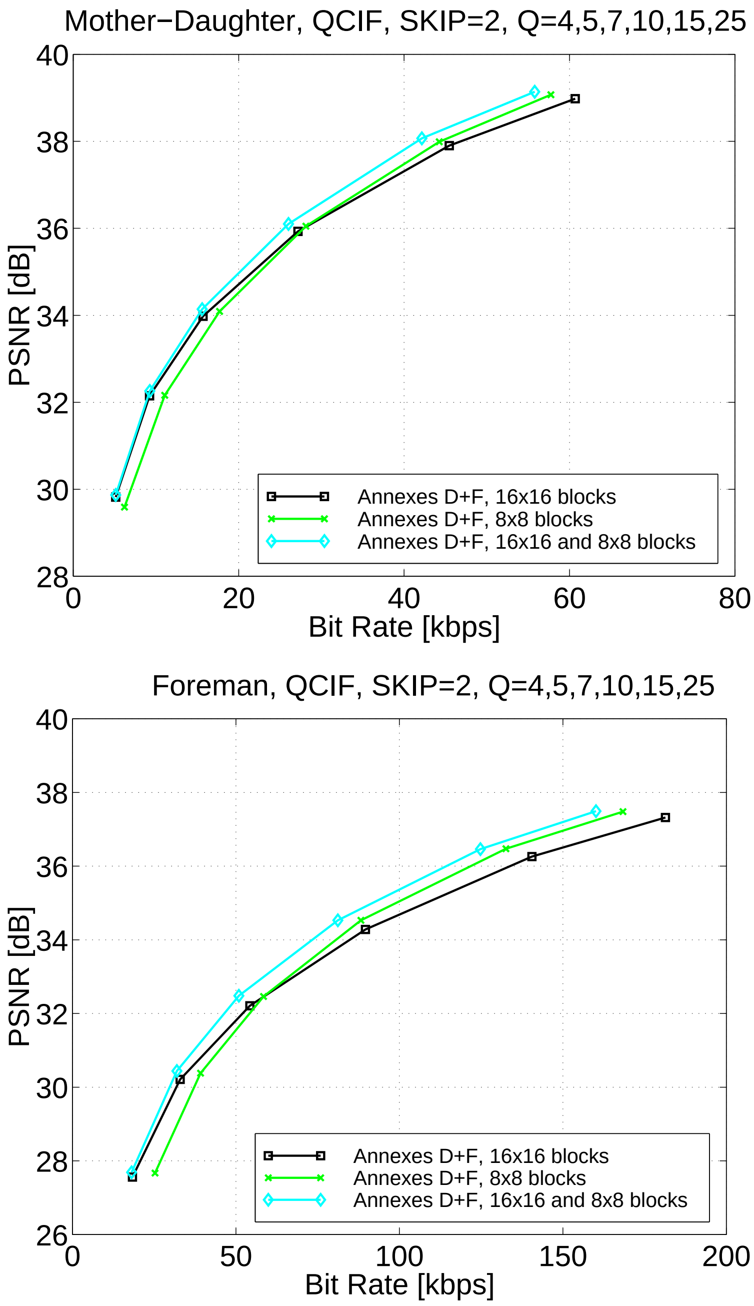

Figure 4: Coding performance for the sequence *Mother & Daughter* (top) and *Foreman* (bottom) when employing variable block sizes.

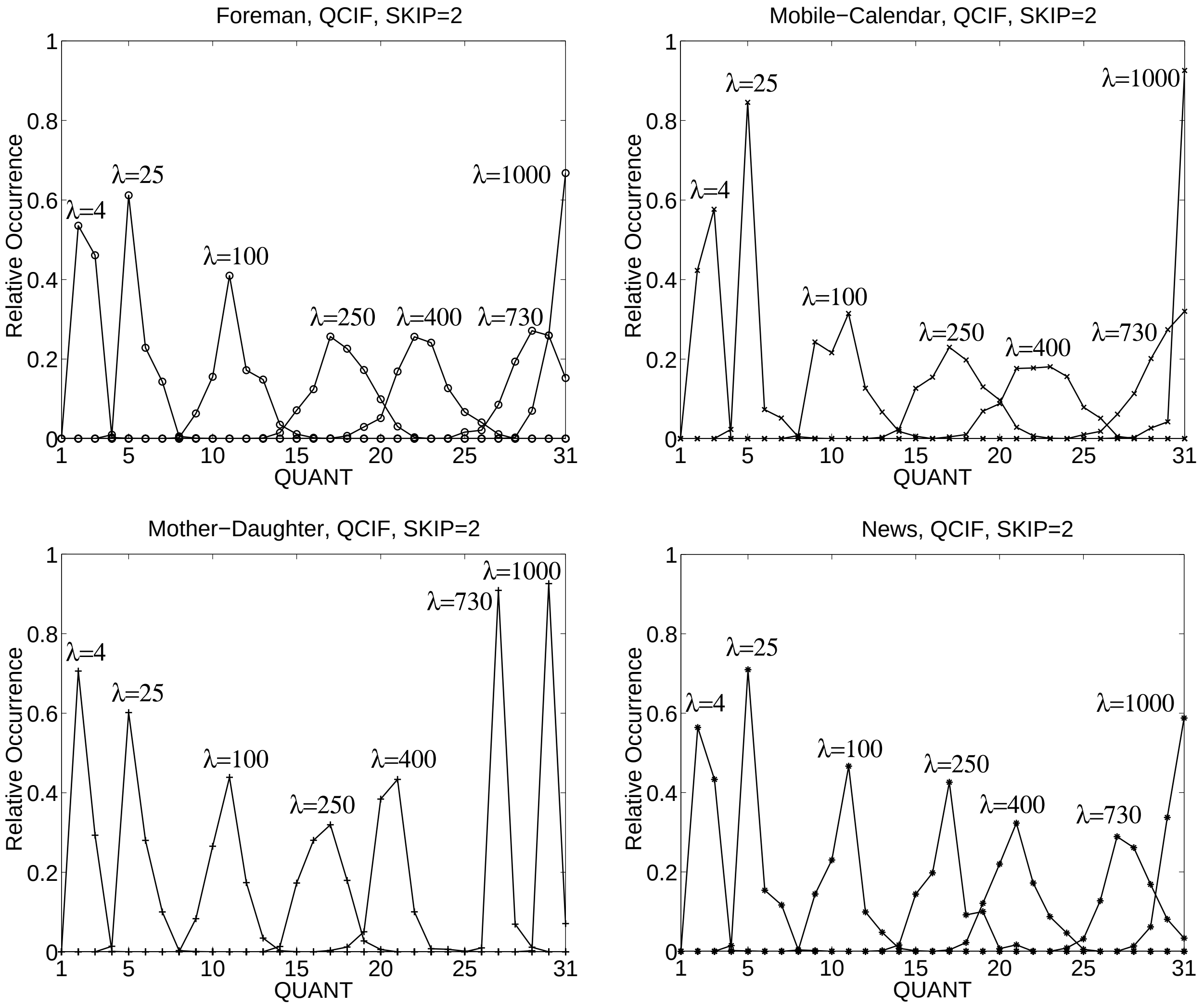

Figure 5: Relative occurrence vs. macroblock QUANT for various Lagrange parameter settings. The relative occurrences of macroblock QUANT values are gathered while coding 100 frames of the video sequences *Foreman* (top left), *Mobile & Calendar* (top right), *Mother & Daughter* (bottom left), and *News* (bottom right).

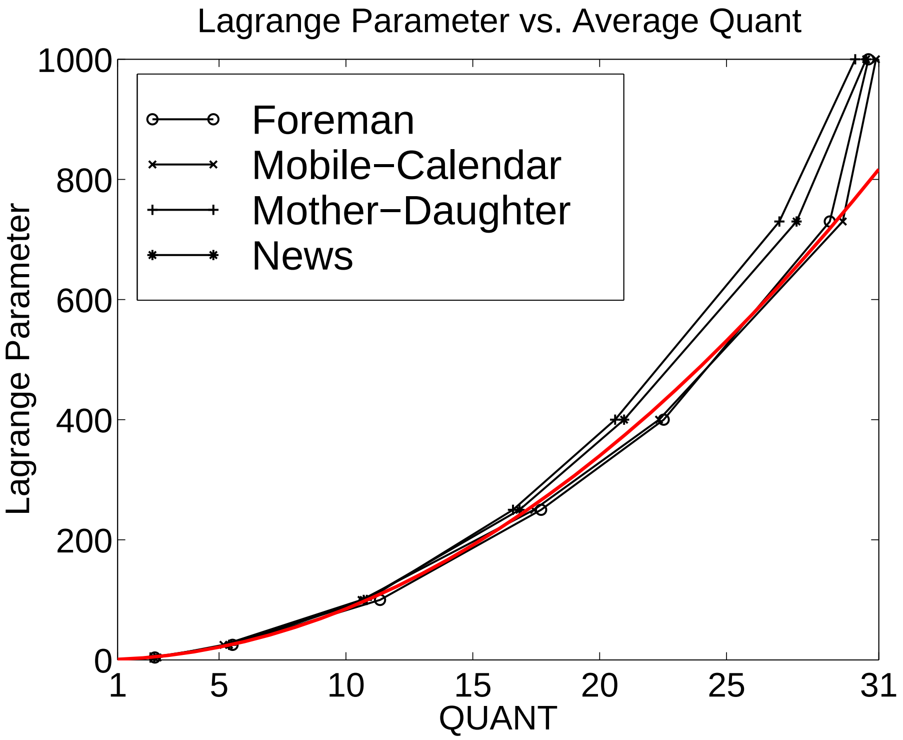

Figure 6: Lagrange parameter λ_{MODE} vs. average macroblock QUANT.

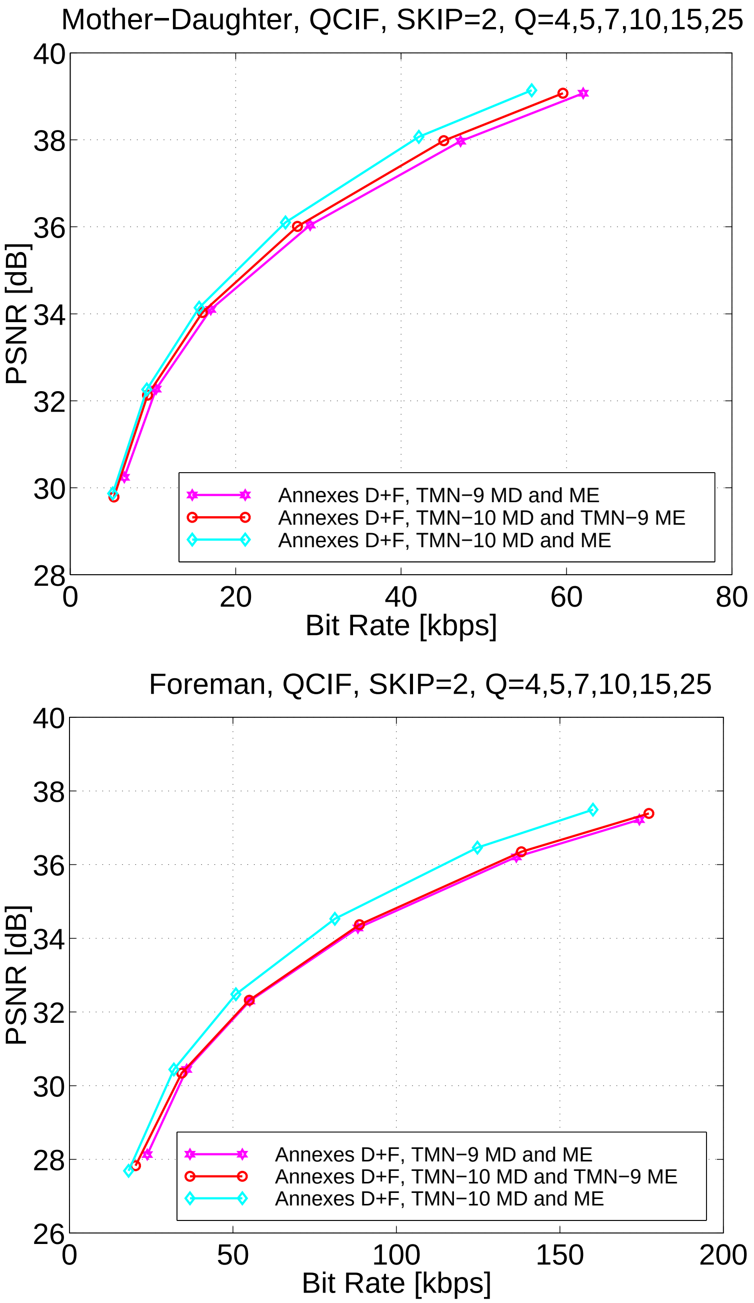

Figure 7: Coding performance for the sequence *Mother & Daughter* (top) and *Foreman* (bottom) when comparing the TMN-9 to the TMN-10 encoding strategy.

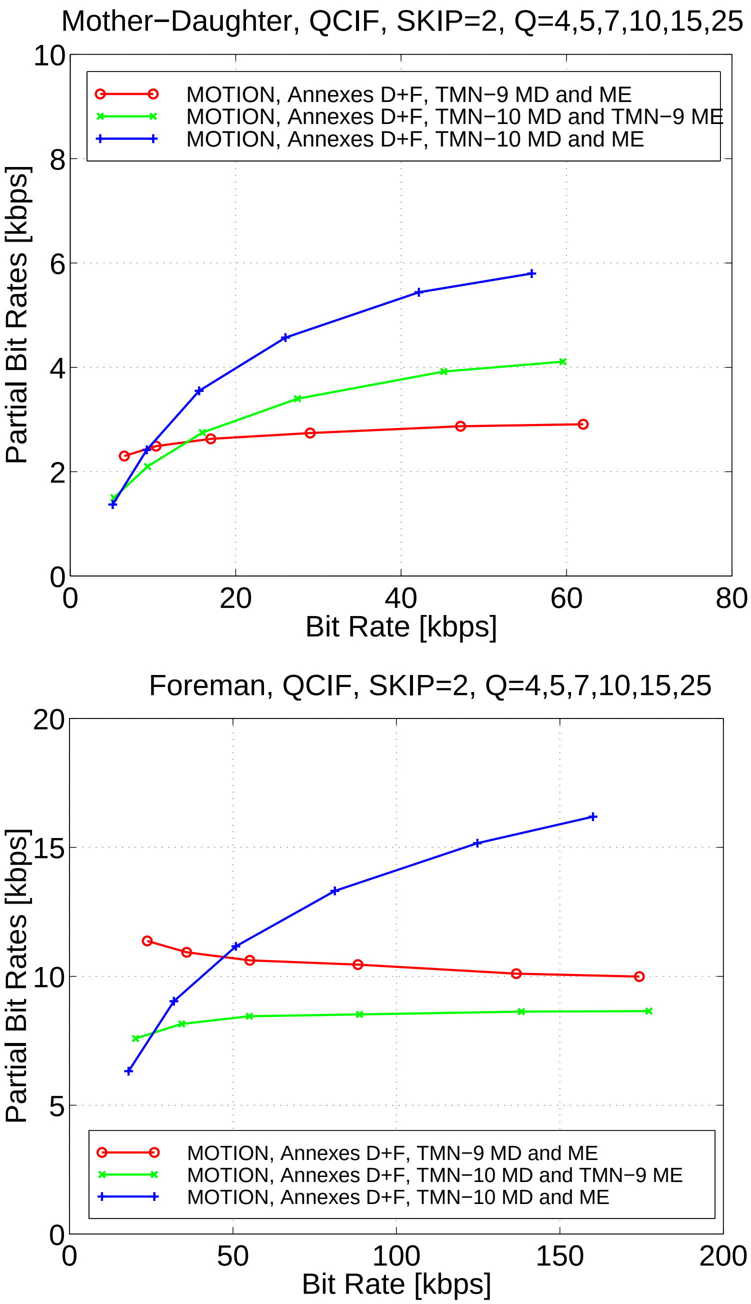

Figure 8: Bit rate partition of MVs vs. bit rate for the sequence *Mother & Daughter* (top) and *Foreman* (bottom) when employing TMN-10 mode decision and motion estimation.