H.263+: Video Coding at Low Bit Rates

Guy Côté, *Student Member, IEEE*, Berna Erol, Michael Gallant, *Student Member, IEEE*, and Faouzi Kossentini, *Member, IEEE*

***Abstract*—In this tutorial paper, we discuss the ITU-T H.263+ (or H.263 Version 2) low-bit-rate video coding standard. We first describe, briefly, the H.263 standard including its optional modes. We then address the 12 new negotiable modes of H.263+. Next, we present experimental results for these modes, based on our public-domain implementation (see our Web site at http://spmg.ece.ubc.ca). Tradeoffs among compression performance, complexity, and memory requirements for the H.263+ optional modes are discussed. Finally, results for mode combinations are presented.**

***Index Terms*— H.263, H.263+, video compression standards, video compression and coding, video conferencing, video telephony.**

I. Introduction

In the past few years, there has been significant interest in digital video applications. Consequently, academia and industry have worked toward developing video compression techniques [1]–[5], and several successful standards have emerged, e.g., ITU-T H.261, H.263, ISO/IEC MPEG-1, and MPEG-2. These standards address a wide range of applications having different requirements in terms of bit rate, picture quality, complexity, error resilience, and delay.

While the demand for digital video communication applications such as videoconferencing, video e-mailing, and video telephony has increased considerably, transmission rates over public switched telephone networks (PSTN) and wireless networks are still very limited. This requires compression performance and channel error robustness levels that cannot be achieved by previous block-based video coding standards such as H.261. Version 1 of the international standard ITU-T H.263, entitled "Video Coding for Low Bit Rate Communications" [6], addresses the above requirements and, as a result, becomes the new low-bit-rate video coding standard.

Although its coding structure is based on that of H.261, H.263 provides better picture quality at low bit rates with little additional complexity. It also includes four optional modes aimed at improving compression performance. H.263 has been adopted in several videophone terminal standards, notably ITU-T H.324 (PSTN), H.320 (ISDN), and H.310 (B-ISDN).

Manuscript received October 26, 1997; revised April 24, 1998. This work was supported by the Natural Sciences and Engineering Research Council of Canada and by AVT Audio Visual Telecommunications Corporation. This paper was recommended by Associate Editor M.-T. Sun.

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, B.C., V6T 1Z4, Canada.

Publisher Item Identifier S 1051-8215(98)06325-3.

H.263 Version 2, also known as H.263+ in the standards community, was officially approved as a standard in January 1998 [7]. H.263+ is an extension of H.263, providing 12 new negotiable modes and additional features. These modes and features improve compression performance, allow the use of scalable bit streams, enhance performance over packet-switched networks, support custom picture size and clock frequency, and provide supplemental display and external usage capabilities.

II. The ITU-T H.263 Standard

The H.263 video standard is based on techniques common to many current video coding standards. In this section, we describe the source coding framework of H.263.

A. Baseline H.263 Video Coding

Fig. 1 shows a block diagram of an H.263 baseline encoder. Motion-compensated prediction first reduces temporal redundancies. Discrete cosine transform (DCT)-based algorithms are then used for encoding the motion-compensated prediction difference frames. The quantized DCT coefficients, motion vectors, and side information are entropy coded using variable-length codes (VLC's).

1) Video Frame Structure: H.263 supports five standardized picture formats: sub-QCIF, QCIF, CIF, 4CIF, and 16CIF. The luminance component of the picture is sampled at these resolutions, while the chrominance components, Cb and Cr, are downsampled by two in both the horizontal and vertical directions. The picture structure is shown in Fig. 2 for the QCIF resolution. Each picture in the input video sequence is divided into macroblocks, consisting of four luminance blocks of 8 pixels $\times$ 8 lines followed by one Cb block and one Cr block, each consisting of 8 pixels $\times$ 8 lines. A group of blocks (GOB) is defined as an integer number of macroblock rows, a number that is dependent on picture resolution. For example, a GOB consists of a single macroblock row at QCIF resolution.

2) Video Coding Tools: H.263 supports interpicture prediction that is based on motion estimation and compensation. The coding mode where temporal prediction is used is called an inter mode. In this mode, only the prediction error frames—the difference between original frames and motion-compensated predicted frames—need be encoded. If temporal prediction is not employed, the corresponding coding mode is called an intra mode.

a) Motion estimation and compensation: Motion-compensated prediction assumes that the pixels within the

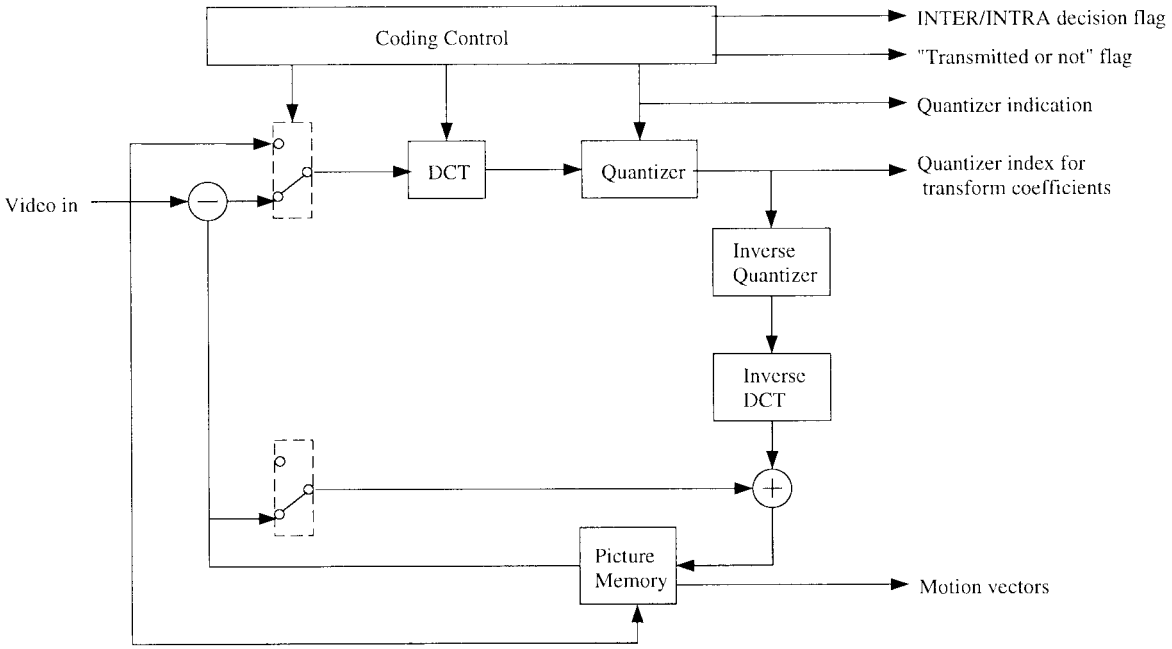


Fig. 1. H.263 video encoder block diagram.

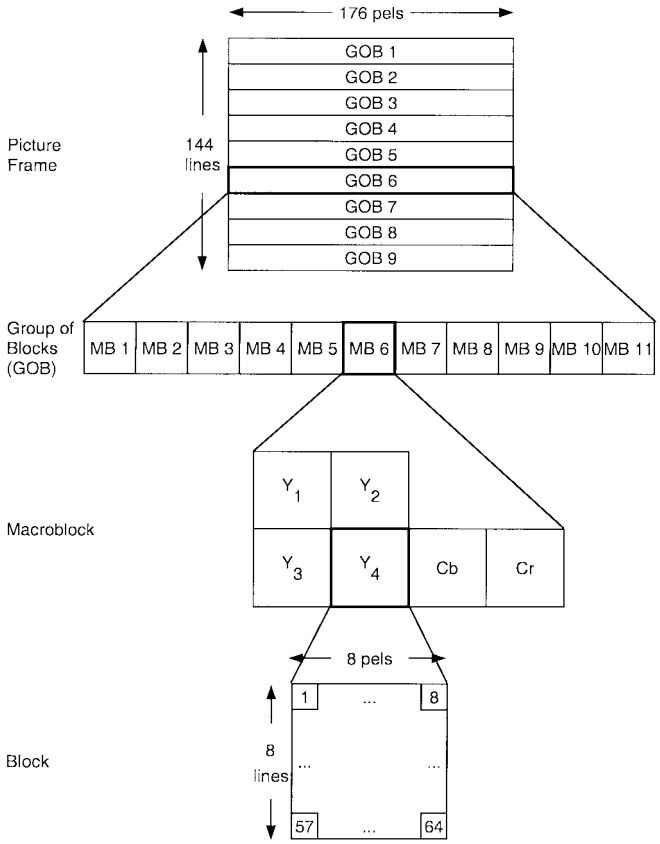


Fig. 2. H.263 picture structure at QCIF resolution.

current picture can be modeled as a translation of those within a previous picture, as shown in Fig. 3. In baseline H.263, each macroblock is predicted from the previous frame. This implies an assumption that each pixel within the macroblock undergoes the same amount of translational

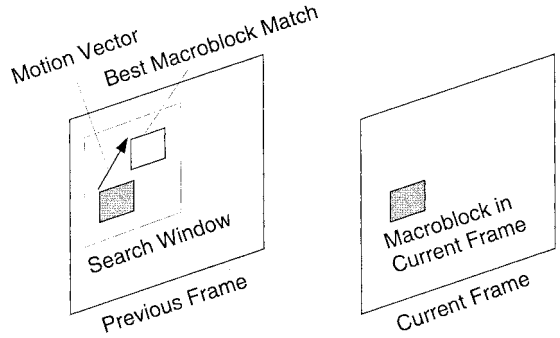


Fig. 3. H.263 source coding algorithm: motion compensation.

motion. This motion information is represented by two-dimensional displacement vectors or motion vectors. Due to the block-based picture representation, many motion estimation algorithms employ block-matching techniques, where the motion vector is obtained by minimizing a cost function measuring the mismatch between a candidate macroblock and the current macroblock. Although several cost measures have been introduced, the most widely used one is the sum-of-absolute-differences (SAD) defined by

$$\mathrm{SAD} = \sum_{k=1}^{16}\sum_{l=1}^{16} \left| B_{i,j}(k,l) - B_{i-u,j-v}(k,l) \right|$$

where $B_{i,j}(k,l)$ represents the (k,l)th pixel of a 16×16 macroblock from the current picture at the spatial location (i,j), and $B_{i-u,j-v}(k,l)$ represents the (k,l)th pixel of a candidate macroblock from a reference picture at the spatial location (i,j) displaced by the vector (u,v). To find the macroblock producing the minimum mismatch error, we need to calculate the SAD at several locations within a search window. The simplest, but the most compute-intensive search

method, known as the full search or exhaustive search method, evaluates the SAD at every possible pixel location in the search area. To lower the computational complexity, several algorithms that restrict the search to a few points have been proposed [8]. In baseline H.263, one motion vector per macroblock is allowed for motion compensation. Both horizontal and vertical components of the motion vectors may be of half pixel accuracy, but their values may lie only in the [−16, 15.5] range, limiting the search window used in motion estimation. A positive value of the horizontal or vertical component of the motion vector represents a macroblock spatially to the right or below the macroblock being predicted, respectively.

b) Transform: The purpose of the 8 × 8 DCT specified by H.263 is to decorrelate the 8 × 8 blocks of original pixels or motion-compensated difference pixels, and to compact their energy into as few coefficients as possible. Besides its relatively high decorrelation and energy compaction capabilities, the 8 × 8 DCT is simple, efficient, and amenable to software and hardware implementations [9]. The most common algorithm for implementing the 8 × 8 DCT is that which consists of eight-point DCT transformation of the rows and the columns, respectively. The 8 × 8 DCT is defined by

$$C_{m,n} = \alpha(m)\beta(n)\sum_{i=1}^{8}\sum_{j=1}^{8} B_{i,j}\cos\left(\frac{\pi(2i+1)m}{16}\right) \cdot \cos\left(\frac{\pi(2j+1)n}{16}\right), \qquad 0 \le m, n \le 7$$

where

$$\alpha(0) = \beta(0) = \sqrt{\tfrac{1}{8}} \quad \text{and} \quad \alpha(m) = \beta(n) = \sqrt{\tfrac{1}{4}} \quad \text{for } 1 \le m, n \le 7.$$

Here, $B_{i,j}$ denotes the (i, j)th pixel of the 8 × 8 original block, and $C_{m,n}$ denotes the coefficients of the 8 × 8 DCT transformed block. The original 8 × 8 block of pixels can be recovered using an 8 × 8 inverse DCT (IDCT) given by

$$B_{i,j} = \sum_{m=1}^{8}\sum_{n=1}^{8} C_{m,n}\alpha(m)\cos\left(\frac{\pi(2m+1)i}{16}\right)\beta(n) \cdot \cos\left(\frac{\pi(2n+1)j}{16}\right), \quad 0 \le i, j \le 7.$$

Although exact reconstruction can be theoretically achieved, it is often not possible using finite-precision arithmetic. While forward DCT errors can be tolerated, inverse DCT errors must meet the H.263 standard if compliance is to be achieved.

c) Quantization: The human viewer is more sensitive to reconstruction errors related to low spatial frequencies than those related to high frequencies [10]. Slow linear changes in intensity or color (low-frequency information) are important to the eye. Quick, high-frequency changes can often not be seen, and may be discarded. For every element position in the DCT output matrix, a corresponding quantization value is computed using the equation

$$C^q_{m,n} = \frac{C_{m,n}}{Q_{m,n}}, \qquad 0 \le m, n \le 7$$

where $C_{m,n}$ is the (m, n)th DCT coefficient and $Q_{m,n}$ is the (m, n)th quantization value. The resulting real numbers are then rounded to their nearest integer values. The net effect is usually a reduced variance between quantized coefficients as compared to the variance between the original DCT coefficients, as well as a reduction of the number of nonzero coefficients.

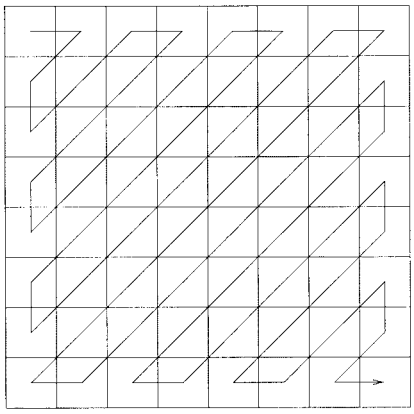


Fig. 4. Zigzag scan pattern to reorder DCT coefficients from low to high frequencies.

In H.263, quantization is performed using the same step size within a macroblock (i.e., using a uniform quantization matrix). Even quantization levels in the range from 2 to 62 are allowed, except for the first coefficient (DC coefficient) of an intra block, which is uniformly quantized using a step size of eight. The quantizers consist of equally spaced reconstruction levels with a dead zone centered at zero. After the quantization process, the reconstructed picture is stored so that it can be later used for prediction of the future picture.

d) Entropy coding: Entropy coding is performed by means of variable-length codes (VLC's). Motion vectors are first predicted by setting their component's values to median values of those of neighboring motion vectors already transmitted: the motion vectors of the macroblocks to the left, above, and above right of the current macroblock. The difference motion vectors are then VLC coded.

Prior to entropy coding, the quantized DCT coefficients are arranged into a one-dimensional array by scanning them in zigzag order. This rearrangement places the DC coefficient first in the array, and the remaining AC coefficients are ordered from low to high frequency. This scan pattern is illustrated in Fig. 4. The rearranged array is coded using a three-dimensional run-length VLC table, representing the triple (LAST, RUN, LEVEL). The symbol RUN is defined as the distance between two nonzero coefficients in the array. The symbol LEVEL is the nonzero value immediately following a sequence of zeros. The symbol LAST replaces the H.261 end-of block flag, where "LAST = 1" means that the current code corresponds to the last coefficient in the coded block. This coding method produces a compact representation of the 8 × 8 DCT coefficients, as a large number of the coefficients are normally quantized to zero and the reordering results (ideally) in the grouping of long runs of consecutive zero values. Other information such as prediction types and quantizer indication is also entropy coded by means of VLC's.

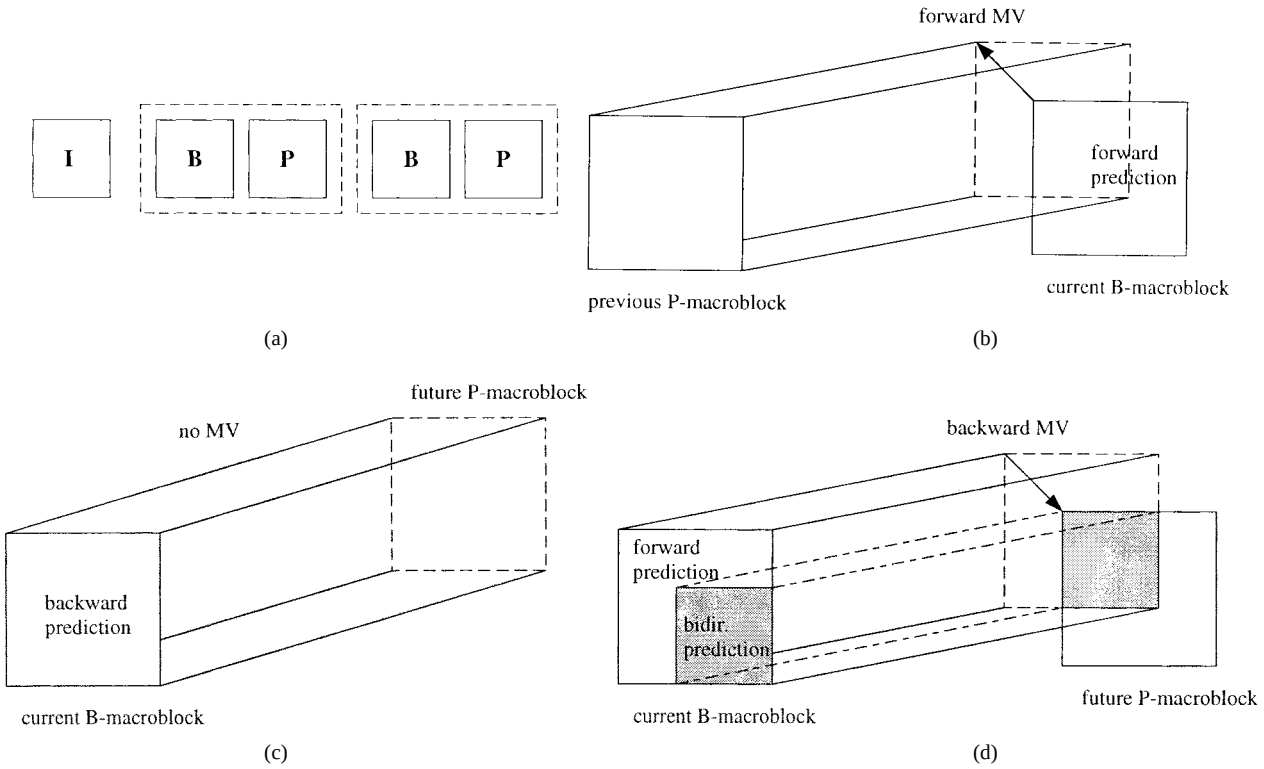


Fig. 5. Improved *PB* frames. (a) Structure. (b) Forward prediction. (c) Backward prediction. (d) Bidirectional prediction.

3) Coding Control: The two switches in Fig. 1 represent the intra/inter mode selection, which is not specified in the standard. Such a selection is made at the macroblock level. The performance of the motion estimation process, usually measured in terms of the associated SAD values, can be used to select the coding mode (intra or inter). If a macroblock does not change significantly with respect to the reference picture, an encoder can also choose not to encode it, and the decoder will simply repeat the macroblock located at the subject macroblock's spatial location in the reference picture.

B. Optional Modes

In addition to the core encoding and decoding algorithms described above, H.263 includes four negotiable advanced coding modes: unrestricted motion vectors, advanced prediction, *PB* frames, and syntax-based arithmetic coding. The first two modes are used to improve inter picture prediction. The *PB*-frames mode improves temporal resolution with little bit rate increase. When the syntax-based arithmetic coding mode is enabled, arithmetic coding replaces the default VLC coding. These optional modes allow developers to trade off between compression performance and complexity. We next provide a brief description of each of these modes. A more detailed description of such modes can be found in [11] and [12].

1) Unrestricted Motion Vector Mode (Annex D): In baseline H.263, motion vectors can only reference pixels that are within the picture area. Because of this, macroblocks at the border of a picture may not be well predicted. When the unrestricted motion vector mode is used, motion vectors can take on values in the range $[-31.5, 31.5]$ instead of $[-16, 15.5]$, and are allowed to point outside the picture boundaries. The longer motion vectors improve coding efficiency for larger picture formats, i.e., 4CIF or 16CIF. Moreover, by allowing motion vectors to point outside the picture, a significant gain is achieved if there is movement along picture edges. This is especially useful in the case of camera movement or background movement.

2) Syntax-Based Arithmetic Coding Mode (Annex E): Baseline H.263 employs variable-length coding as a means of entropy coding. In this mode, syntax-based arithmetic coding is used. Since VLC and arithmetic coding are both lossless coding schemes, the resulting picture quality is not affected, yet the bit rate can be reduced by approximately 5% due to the more efficient arithmetic codes. It is worth noting that use of this annex is not widespread.

3) Advanced Prediction Mode (Annex F): This mode allows for the use of four motion vectors per macroblock, one for each of the four 8×8 luminance blocks. Furthermore, overlapped block motion compensation is used for the luminance macroblocks, and motion vectors are allowed to point outside the picture as in the unrestricted motion vector mode. Use of this mode improves inter picture prediction, and yields a significant improvement in subjective picture quality for the same bit rate by reducing blocking artifacts.

4) PB-Frames Mode (Annex G): In this mode, the frame structure consists of a *P* picture and a *B* picture, as illustrated in Fig. 5(a). The quantized DCT coefficients of the *B* and *P* pictures are interleaved at the macroblock layer such that a *P*-picture macroblock is immediately followed by a *B*-picture macroblock. Therefore, the maximum number of blocks transmitted at the macroblock layer is 12 rather than 6. The *P* picture is forward predicted from the previously decoded *P*

picture. The *B* picture is bidirectionally predicted from the previously decoded *P* picture and the *P* picture currently being decoded. The forward and backward motion vectors for a *B* macroblock are calculated by scaling the motion vector from the current *P*-picture macroblock using the temporal resolution of the *P* and *B* pictures with respect to the previous *P* picture. If this motion vector does not yield a good prediction, it can be enhanced by a delta vector. The delta vector is obtained by performing motion estimation, within a small search window, around the calculated motion vectors.

When decoding a *PB*-frame macroblock, the *P* macroblock is reconstructed first, followed by the *B* macroblock since the information from the *P* macroblock is needed for *B*-macroblock prediction. When using the *PB*-frames mode, the picture rate can be doubled without a significant increase in bit rate.

III. The ITU-T H.263+ Standard

The objective of H.263+ is to broaden the range of applications and to improve compression efficiency. H.263+, or H.263 version 2, is backward compatible with H.263. Not only is this critical due to the large number of video applications currently using the H.263 standard, but it is also required by ITU-T rules.

H.263+ offers many improvements over H.263. It allows the use of a wide range of custom source formats, as opposed to H.263, wherein only five video source formats defining picture size, picture shape, and clock frequency can be used. This added flexibility opens H.263+ to a broader range of video scenes and applications, such as wide format pictures, resizeable computer windows, and higher refresh rates. Moreover, picture size, aspect ratio, and clock frequency can be specified as part of the H.263+ bit stream. Another major improvement of H.263+ over H.263 is scalability, which can improve the delivery of video information in error-prone, packet-lossy, or heterogeneous environments by allowing multiple display rates, bit rates, and resolutions to be available at the decoder. Furthermore, picture segment[1] dependencies may be limited, likely reducing error propagation.

A. H.263+ Optional Modes

Next, we describe each of the 12 new optional coding modes of the H.263+ video coding standard, including the modification of H.263's unrestricted motion vector mode when used within an H.263+ framework.

1) Unrestricted Motion Vector Mode (Annex D): The definition of the unrestricted motion vector mode in H.263+ is different from that of H.263. When this mode is employed within an H.263+ framework, new reversible VLC's (RVLC's) are used for encoding the difference motion vectors. These codes are single valued, as opposed to the earlier H.263 VLC's which were double valued. The double-valued codes were not popular due to limitations in their extendibility, and also to their high implementation cost. Reversible VLC's are easy to implement as a simple state machine can be used to generate and decode them.

More importantly, reversible VLC's can be used to increase resilience to channel errors. The idea behind RVLC's is that decoding can be performed by processing the received motion vector part of the bit stream in the forward and reverse directions. If an error is detected while decoding in the forward direction, motion vector data are not completely lost as the decoder can proceed in the reverse direction; this improves error resilience of the bit stream [13].[2] Furthermore, the motion vector range is extended to up to $[-256, +255.5]$ depending on the picture size, as depicted in Table I. This is very useful given the wide range of new picture formats available in H.263+.

2) Advanced Intra Coding Mode (Annex I): This mode improves compression performance when coding intra macroblocks. In this mode, inter block prediction from neighboring intra coded blocks, a modified inverse quantization of intra DCT coefficients, and a separate VLC table for intra coded coefficients are employed. Block prediction is performed using data from the same luminance or chrominance components (Y, Cr, or Cb). As illustrated in Fig. 6, one of three different prediction options can be signaled: DC only, vertical DC and AC, or horizontal DC and AC. In the DC only option, only the DC coefficient is predicted, usually from both the block above and the block to the left, unless one of these blocks is not in the same picture segment or is not an intra block. In the vertical DC and AC option, the DC and first row of AC coefficients are vertically predicted from those of the block above. Finally, in the horizontal DC and AC option, the DC and first column of AC coefficients are horizontally predicted from those of the

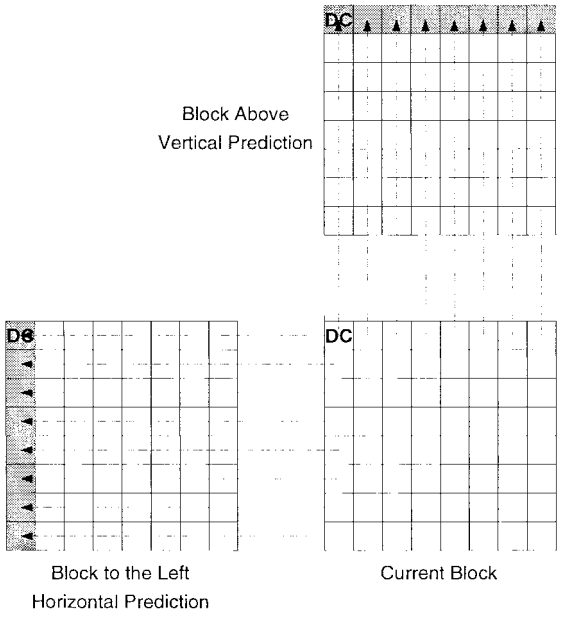


Fig. 6. Neighboring blocks used for intra prediction in the advanced intra coding mode.

[1] A picture segment is defined as a slice or any number of GOB's preceded by a GOB header.

[2] To exploit the full error resilience potential of RVLC's, the motion vector bits should be blocked into one stream for each video frame, concatenating a large number of RLVC's. This can be performed by data partitioning, which is currently being proposed in H.263++.

TABLE I
MOTION VECTOR RANGE IN H.263+'S UNRESTRICTED MOTION VECTOR RANGE MODE

Picture width	Horizontal motion vector range	Picture height	Vertical motion vector range
4, ..., 352	[-32, 31.5]	4, ..., 288	[-32, 31.5]
356, ..., 704	[-64, 63.5]	292, ..., 576	[-64, 63.5]
708, ..., 1408	[-128, 127.5]	292, ..., 576	[-64, 63.5]
1412, ..., 2048	[-256, 255.5]	580, ..., 1152	[-128, 127.5]

block to the left. The option that yields the best prediction is applied to all blocks of the subject intra macroblock.

The difference coefficients, obtained by subtracting the predicted DCT coefficients from the original ones, are then quantized and scanned differently, depending on the selected prediction option. Three scanning patterns are used: the basic zigzag scan for DC only prediction, the alternate-vertical scan (as in MPEG-2) for horizontally predicted blocks, or the alternate-horizontal scan for vertically predicted blocks. The main part of the standard employs the same VLC table for coding all quantized coefficients. However, this table is designed for inter macroblocks and is not very effective for coding intra macroblocks. In intra macroblocks, larger coefficients with smaller runs of zeros are more common. Thus, the advanced intra coding mode employs a new VLC table for encoding the quantized coefficients, a table that is optimized to global statistics of intra macroblocks.

3) Deblocking Filter Mode (Annex J): This mode introduces a deblocking filter inside the coding loop. Unlike in postfiltering, predicted pictures are computed based on filtered versions of the previous ones. A filter is applied to the edge boundaries of the four luminance and two chrominance 8×8 blocks. The filter is applied to a window of four edge pixels in the horizontal direction, and it is then similarly applied in the vertical direction. The weight of the filter's coefficients depend on the quantizer step size for a given macroblock, where stronger coefficients are used for a coarser quantizer. This mode also allows the use of four motion vectors per macroblock, as specified in the advanced prediction mode of H.263, and also allows motion vectors to point outside picture boundaries, as in the unrestricted motion vector mode. The above techniques, as well as filtering, result in better prediction and a reduction in blocking artifacts. The computationally expensive overlapping motion compensation operation of the advanced prediction mode is not used here in order to keep the additional complexity of this mode minimal.

4) Slice Structured Mode (Annex K): A slice structure, instead of a GOB structure, is employed in this mode. This allows the subdivision of the picture into segments containing variable numbers of macroblocks. The slice structure consists of a slice header followed by consecutive complete macroblocks. Two additional submodes can be signaled to reflect the order of transmission, sequential or arbitrary, and the shape of the slices, rectangular or not. These add flexibility to the slice structure so that it can be designed for different environments and applications. For example, rectangular slices can be used to subdivide a picture into rectangular regions of interest for region-based coding. The slice header locations within the bit stream act as resynchronization points, which help the decoder recover from bit errors and packet losses. They also allow slices to be decoded in an arbitrary order.

5) Supplemental Enhancement Information Mode (Annex L): In this mode, supplemental information is included in the bit stream in order to offer display capabilities within the coding framework. This supplemental information includes support for picture freeze, picture snapshot, video segmentation, progressive refinement, and chroma keying. These added functionalities are externally negotiated at the system layer (using H.245 for example) to ensure picture synchronization.

The picture freeze option allows the encoder to signal a complete or partial freeze of a picture. Rectangular areas of a picture can be frozen while the rest of the picture is still being updated. A picture freeze release code is explicitly sent to the decoder. The picture snapshot option allows part of or the full picture to be used as a still image snapshot by an external application. When video subsequences can be used by an external application, such can be signaled by the video segmentation option of this mode. The progressive refinement option signals to the decoder that the following pictures represent a refinement in quality of the subject picture, as opposed to pictures at different times. The chroma keying option indicates that transparent or semitransparent pixels can be employed during the video decoding process. When set on, transparent pixels are not displayed. Instead, a background picture that is externally controlled is displayed.

All of the above options are aimed at providing decoder supporting features and functionalities within the video bit stream. For example, such options will facilitate interoperability between different applications within the context of windows-based environments.

6) Improved PB-Frames Mode (Annex M): This mode is an enhanced version of the H.263 *PB*-frames mode. The main difference is that the H.263 *PB*-frames mode allows only bidirectional prediction to predict *B* pictures in a *PB* frame, whereas the improved *PB*-frames mode permits forward, backward, and bidirectional prediction as illustrated in Fig. 5. Bidirectional prediction methods, as illustrated in Fig. 5(d), are the same in both modes, except that, in the improved *PB*-frames mode, no delta vector is transmitted. In forward prediction, as shown in Fig. 5(b), the *B* macroblock is predicted from the previous *P* macroblock, and a separate motion vector is then transmitted. In backward prediction, as illustrated in Fig. 5(c), the predicted macroblock is equal to the future *P* macroblock, and therefore no motion vector is transmitted. Use of the additional forward and backward predictors makes the improved *PB* frames less susceptible to significant changes that may occur between pictures.

7) Reference Picture Selection Mode (Annex N): In H.263, a picture is predicted from the previous picture. If a part of the subject picture is lost due to channel errors or packet loss, the quality of future pictures can be severely degraded. Using this mode, it is possible to select the reference picture for prediction in order to suppress temporal error propagation due to inter coding. Multiple pictures must be stored at the decoder, and the encoder should signal the necessary amount of additional picture memory by external means. The information which

specifies the selected picture for prediction is included in the encoded bit stream.

If a back-channel is employed, two back-channel mode switches define four messaging methods (NEITHER, ACK, NACK, and ACK+NACK) that the encoder and decoder employ to determine which picture segment will be used for prediction. For example, a NACK sent to the encoder from the decoder signals that a given picture has been degraded by errors. Thus, the encoder may choose not to use this picture for future prediction, and instead employ a different, unaffected, reference picture. This mode reduces error propagation, thus maintaining good picture reproduction quality in error-prone environments.

8) Temporal, SNR, and Spatial Scalability Mode (Annex O): This mode specifies syntax to support temporal, SNR, and spatial scalability capabilities. Scalability is a desirable property for error-prone and heterogeneous environments. It implies that the encoder's output bit stream can be manipulated any time after it has been generated. This property is desirable in order to counter limitations such as constraints on bit rate, display resolution, network throughput, and decoder complexity. In multipoint and broadcast video applications, such constraints cannot be foreseen at the time of encoding.

Temporal scalability provides a mechanism for enhancing perceptual quality by increasing the picture display rate. This is achieved via bidirectionally predicted pictures, inserted between anchor picture pairs and predicted from either one or both of these anchor pictures, as illustrated in Fig. 7(a). Thus, for the same quantization level, *B* pictures yield increased compression as compared to forward predicted *P* pictures. *B* pictures are not used as anchor pictures, i.e., other pictures are never predicted from them. Therefore, they can be discarded without impacting picture quality of future pictures; hence, the name temporal scalability. Note that, while *B* pictures improve compression performance as compared to *P* pictures, they increase encoder complexity and memory requirements and introduce additional delays.

Spatial scalability and SNR scalability are closely related, the only difference being the increased spatial resolution provided by spatial scalability. An example of SNR scalable pictures is shown in Fig. 7(b). SNR scalability implies the creation of multirate bit streams. It allows for the recovery of coding error, or the difference, between an original picture and its reconstruction. This is achieved by using a finer quantizer to encode the difference picture in an enhancement layer. This additional information increases the SNR of the overall reproduced picture; hence, the name SNR scalability.

Spatial scalability allows for the creation of multiresolution bit streams to meet varying display requirements/constraints for a wide range of clients. A spatial scalable structure is illustrated in Fig. 7(c). It is essentially the same as in SNR scalability, except that a spatial enhancement layer here attempts to recover the coding loss between an upsampled version of the reconstructed reference layer picture and a higher resolution version of the original picture. For example, if the reference layer has a QCIF resolution, and the enhancement layer has a CIF resolution, the reference layer picture must be scaled accordingly such that the enhancement layer picture

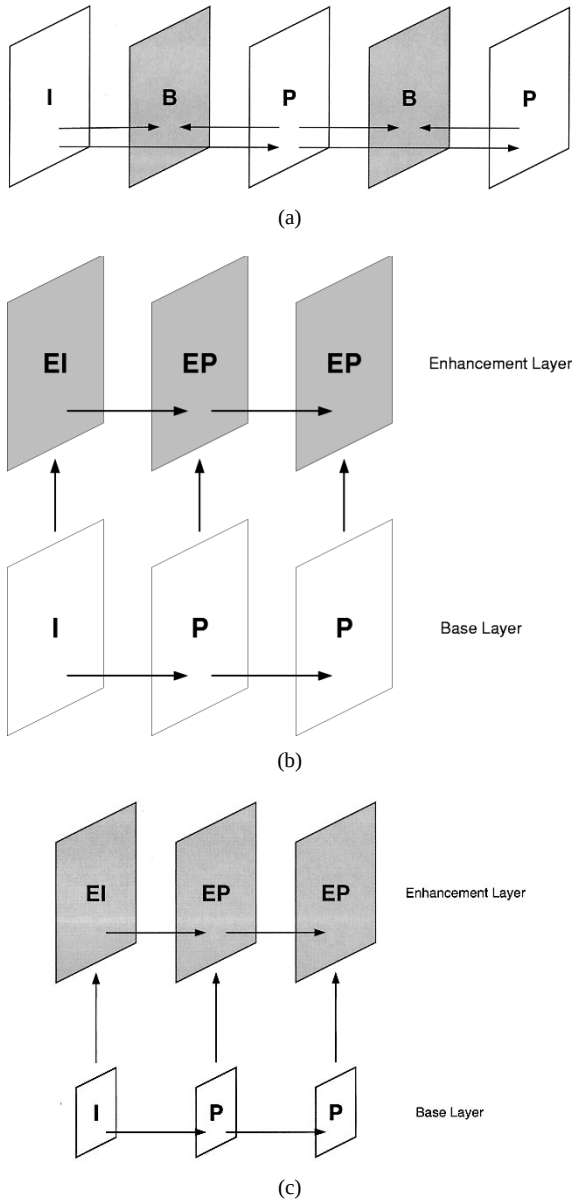


Fig. 7. Illustration of scalability features. (a) Temporal. (b) SNR. (c) Spatial.

can be appropriately predicted from it. The standard allows the resolution to be increased by a factor of 2 in the vertical only, horizontal only, or both the vertical and horizontal directions for a single enhancement layer. There can be multiple enhancement layers, each increasing picture resolution over that of the previous layer. The interpolation filters used to upsample the reference layer picture are explicitly defined in the standard. Aside from the upsampling process from the reference to the enhancement layer, the processing and syntax of a spatially scaled picture are identical to those of an SNR scaled picture.

In either SNR or spatial scalability, the enhancement layer pictures are referred to as *EI* or *EP* pictures. If the enhancement layer picture is upward predicted, from a picture in the reference layer, then the enhancement layer picture is referred to as an enhancement-*I* (*EI*) picture. In some cases, when reference layer pictures are coarsely represented, over coding of static parts of the picture can occur in the enhancement

layer, requiring an unnecessarily excessive bit rate. To avoid this problem, forward prediction is permitted in the enhancement layer. A picture that can be forward predicted from a previous enhancement layer picture or upward predicted from the reference layer picture is referred to as an enhancement-*P* (*EP*) picture. Note that computing the average of the upward and forward predicted pictures can provide a bidirectional prediction option for *EP* pictures. For both *EI* and *EP* pictures, upward prediction from the reference layer picture implies that no motion vectors are required. In the case of forward prediction for *EP* pictures, motion vectors are required.

9) Reference Picture Resampling Mode (Annex P): This mode describes an algorithm to warp the reference picture prior to its use for prediction. It can be useful for resampling a reference picture having a different source format than the picture being predicted. It can also be used for global motion estimation, or estimation of rotating motion, by warping the shape, size, and location of the reference picture. The syntax includes warping parameters to be used as well as a resampling algorithm.

The simplest level of operation for the reference picture resampling mode is an implicit factor of 4 resampling as only an FIR filter needs to be applied for the upsampling and downsampling processes. In this case, no additional signaling overhead is required as its use is understood when the size of a new picture (indicated in the picture header) is different from that of the previous picture.

10) Reduced Resolution Update Mode (Annex Q): This mode is most useful for highly active motion scenes with detailed backgrounds. It allows the encoder to send update information for a picture encoded at a lower resolution, while still maintaining a higher resolution for the reference picture, to create a final image at the higher resolution. The syntax is the same as the baseline syntax, but interpretation of the semantics is different. The dimensions of the macroblocks are doubled, so the macroblock data size is one-quarter of what it would have been without this mode enabled. Therefore, motion vectors must be doubled in both dimensions. To produce the final picture, the macroblock is upsampled to the intended resolution. After upsampling, the full-resolution texture video picture is added to the motion-compensated picture to create the full-resolution picture for future reference.

This mode is most useful in the case of movement over picture boundaries and motion of large objects. The encoder should have some means of detecting active scenes. It can then choose to enable the reduced resolution update mode for the corresponding pictures.

11) Independently Segmented Decoding Mode (Annex R): In this mode, picture segment boundaries are treated as picture boundaries in the sense that no data dependencies across the segment boundaries are allowed. This includes estimation of motion vectors and texture operations across picture boundaries. Use of this mode limits the propagation of errors, thus providing enhanced error resilience and recovery capabilities. This mode can be better used with slice structures, where a slice can be sized to match a specific packet size, for example.

12) Alternative Inter VLC Mode (Annex S): The intra VLC table designed for encoding quantized intra DCT coefficients in the advanced intra coding mode can be used for inter block coding when this mode is enabled. Large quantized coefficients and small runs of zeros, typically present in intra blocks, become more frequent in inter blocks when small quantizer step sizes are used. When bit savings are obtained, and the use of the intra quantized DCT coefficient table can be detected at the decoder, the encoder will use the intra table. The decoder will first try to decode the quantized coefficients using the inter table. If this results in addressing coefficients beyond the 64 coefficients of the 8×8 block, the decoder will use the intra table.

13) Modified Quantization Mode (Annex T): The modified quantization mode includes three features. First, it allows rate control methods more flexibility in changing the quantizer at the macroblock layer. Second, it enhances chrominance quality by specifying a finer chrominance quantizer step size. Third, it improves picture quality by extending the range of representable quantized DCT coefficients.

In H.263, it is possible to modify the quantizer value at the macroblock level. However, only a small adjustment (± 1 or ± 2) in the value of the most recent quantizer is permitted. The modified quantization mode allows the modification of the quantizer to any value.

In H.263, the luminance and chrominance quantizers are the same. The modified quantization mode increases chrominance picture quality significantly by using a smaller quantizer step size for the chrominance blocks relative to the luminance blocks.

In H.263, when a quantizer smaller than eight is employed, quantized coefficients exceeding the representable range of $[-127, +127]$ are clipped. The modified quantization mode allows coefficients that are outside the range of $[-127, +127]$ to be represented. Therefore, when a very fine quantizer step size is selected, an increase in luminance quality can be obtained.

IV. Test Model Rate Control Methods

The latest version of the H.263+ Test Model, TMN-8 [14], describes two rate control algorithms suitable for low delay videophone applications. Both methods use a buffer regulation scheme in which a target bit rate is chosen and pictures are skipped until the buffer reaches a limit below the number of bits required to transmit the next picture. Since encoding delays are directly related to buffer fullness, large variations in buffer content will produce undesirable variable delays.

The rate control methods try to achieve a target bit rate by changing the macroblock quantizer. The most recent rate control method, also described in [15], is based on a model that chooses an "optimal" quantizer for every macroblock in a given picture. First, the variances of all macroblocks in the motion-compensated picture are calculated. Based on these variances, and the remaining bits available for encoding the current picture, model parameters are updated. These parameters are then used to find an "optimal" quantizer for each macroblock. One of the model parameters allows for the weighting of macroblocks based on perceptual importance. The test model describes a simple method to calculate this

parameter where a macroblock with high spatial activity (higher variance) is assigned a finer quantizer.

The alternate rate control method, described in the TMN-5, TMN-6, and TMN-7 test models, uses a simpler technique for adapting the quantizer. In this method, the quantizer is changed every macroblock row according to the bits remaining for the current picture. This method is simpler to implement than the one previously described, but it also does not provide accurate quantizer selection making it less effective.

V. Experimental Results

The results discussed in this section are based on our implementation of H.263+ [16]. The results illustrate the tradeoffs among compression performance, complexity, encoding/decoding speed, and memory requirements of each of the implemented modes: advanced intra coding mode (Annex I), deblocking filter mode (Annex J), improved *PB*-frames mode (Annex M), temporal, SNR, and spatial scalability mode (Annex O), alternative inter VLC mode (Annex S), and modified quantization mode (Annex T). The TMN-8 rate control methods are also compared in this section. Error resilience/recovery modes (slice structure mode, independently segmented mode, and reference picture selection mode) have already been tested in a packet-lossy environment, and detailed discussions and results can be found in the accompanying paper [17].

The average peak signal-to-noise ratio (PSNR) is used as a distortion measure, and is given by

$$\text{PSNR} = 10 \log \frac{1}{M} \sum_{n=1}^{M} \frac{255^2}{(o_n - r_n)^2}$$

where M is the number of samples and o_n and r_n are the amplitudes of the original and reconstructed pictures, respectively. The average PSNR of all encoded pictures is here used as a measure of objective quality. All of the results represent pictures 0–299 of test sequences having QCIF resolution. Unless otherwise specified, rate control strategies are not employed. Instead, a quantizer step size is fixed for an entire sequence. By selecting different values for the quantizer step size, results are obtained for several bit rates.

The H.263+ TMN-8 model specifies two full-pixel accuracy motion estimation techniques: the conventional full-search technique and a fast-search technique. Unless otherwise specified, all results are obtained using the fast-search motion estimation implementation. Performance of this fast-search algorithm, described in [18], is close to that of the full-search algorithm for a given bit rate. This is illustrated in Table II for the two video sequences *Foreman* and *Akiyo*, coded at 10 frames/s (fps), where the new TMN-8 rate control was used and no optional modes were enabled. For the very active video sequence *Foreman*, the cost of using fast search is approximately 0.2 dB, but for typical videophone sequences such as *Akiyo*, the PSNR levels are essentially the same.

TABLE II
Performance Improvements in PSNR for TMN-8's Full-Search Motion Estimation over Its Fast Search Alternative

Sequence	8 kbps	16 kbps	32 kbps	64 kbps	128 kbps
Foreman		+0.27	+0.15	+0.27	+0.23
Akiyo	-0.05	+0.02	+0.05	+0.01	

A. Advanced Intra Coding Mode

This mode significantly improves compression of intra macroblocks. Prediction lowers the number of bits required to represent the quantized DCT coefficients, while quantization without a dead zone improves the picture reproduction quality. This is illustrated in Fig. 8, which presents coding results of the first intra pictures (i.e., where all of the macroblocks are intra coded) for the Y component of the video sequences *News* and *Akiyo*. Compression improvements of 15–25% are achieved. However, the advanced intra coding mode only improves compression performance of intra coded macroblocks. Thus, negligible compression improvements are achieved for low-activity video sequences, where most macroblocks are inter coded.

Based on our implementation, the associated encoding time increases by 5% on average, due to the prediction method selection operations. This mode requires slightly more memory to store the reconstructed DCT coefficients needed for intra prediction. The increase in decoding time is negligible as only a few additions are required to predict an intra coded macroblock.

B. Deblocking Filter Mode

The deblocking filter mode improves subjective quality by removing blocking and mosquito artifacts common to block-based video coding at low bit rates. Many applications make use of a postfilter to reduce these artifacts. This postfilter is usually present at the decoder, and is outside the coding loop. Therefore, prediction is not based on the postfiltered version of the picture. In our simulations, we used the postfilter described in the TMN-8 test model [14] for comparison with the deblocking filter. Objective results of using the deblocking filter alone are presented in Table IV. As seen in this table, the filtering process may decrease PSNR values. However, the subjective quality is usually improved significantly, as shown in Fig. 9. In the figure, results are shown for the sequence *Foreman* decoded using the deblocking filter alone, TMN-8 postfilter alone, and both the deblocking filter and postfilter at 24 kbits/s and 10 fps. The reconstructed picture for frame number 75 is shown. The deblocking filter alone reduces blocking artifacts significantly, mainly due to the use of four motion vectors per macroblock. The filtering process provides smoothing, further improving subjective quality. The effects of the postfilter are less noticeable, and adding the postfilter may actually result in blurriness. Therefore, the use of the deblocking filter alone is usually sufficient.

Like the advanced prediction mode of H.263, the deblocking filter mode involves using four motion vectors per macroblock. This requires additional motion estimation, increasing the computational load and resulting in a 5–10% additional encoding time. However, if the advanced prediction mode is employed, the additional computational requirements associated with the deblocking filter mode are quite small. The advanced

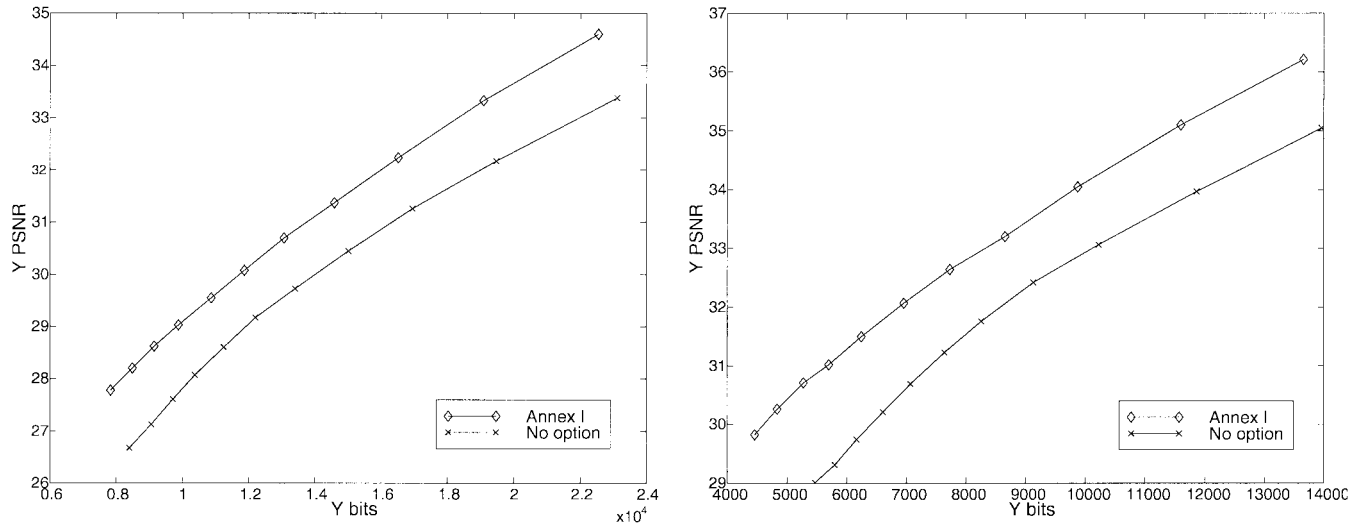


Fig. 8. Advanced intra coding rate-distortion performance for (a) *News* and (b) *Akiyo*.

Fig. 9. Deblocking filter and TMN-8 postfilter: subjective results.

prediction mode already involves using four motion vectors, and only some additional filtering operations are required.

C. Improved PB-Frames Mode

The *PB*-frames mode of H.263 can double the picture rate without significantly increasing the bit rate. The increase in bit rate is slight due mainly to bits saved by coarser quantization of *B* macroblocks. While this causes the *B* picture to have a lower quality than the *P* picture, the increased temporal resolution results in much better overall subjective quality.

The *PB*-frames mode provides good compression performance levels, especially for low motion video sequences. However, since only bidirectional prediction is used for the *B* picture of the *PB* frame, when irregular motion is present

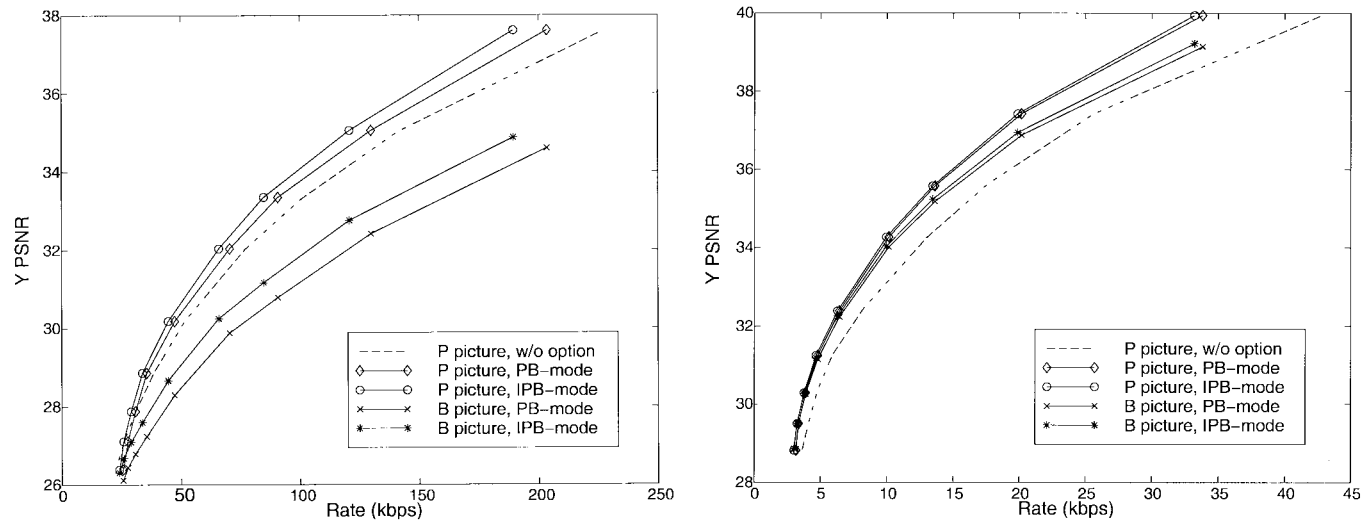


Fig. 10. Improved *PB*-frames mode: rate-distortion performance for (a) *Foreman* and (b) *Akiyo*.

in the video sequence, quality of the *B* picture decreases considerably. The improved *PB*-frames mode of H.263+ addresses this problem by allowing forward only *or* backward only prediction, in addition to bidirectional prediction, of *B* macroblocks. Since the *B* and *P* pictures are encoded at the same time, reconstructed *P* macroblocks to the right of and below the current *P* macroblock[3] are not available for backward prediction of the current *B* macroblock. Although causal *P* macroblocks are still available for backward prediction, the H.263+ standard requires that only the current *P* macroblock may be used to predict the current *B* macroblock. Therefore, backward motion vectors are not used. While this type of backward prediction is not as effective as that of true *B* pictures, it does make the H.263+ *IPB*-frames mode *B* pictures more robust than those of the H.263 *PB*-frames mode with respect to scene changes in video sequences.

Our simulation results show that a significant improvement in PSNR is achieved as compared to the H.263 *PB*-frames mode when an active video sequence is coded. This is illustrated in Fig. 10(a). The figure shows the PSNR gains achieved by enabling the H.263 *PB*-frames mode or the H.263+ improved *PB*-frames mode, as compared to using only the H.263 baseline coder, for the luminance component of the active video sequence *Foreman* at 10 fps. Also, using the H.263+ improved *PB*-frames mode instead of the H.263 *PB*-frames mode, the PSNR of *B* pictures is increased by approximately 0.5 dB for a given bit rate. On the other hand, the PSNR gain over the H.263 *PB*-frames mode is smaller for video sequences that have moderate motion. A good example is the low motion video sequence *Akiyo*, as shown in Fig. 10(b). Clearly, there is only a small gain in PSNR for both the *P* and *B* pictures.

Both the H.263 *PB*-frames and H.263+ improved *PB*-frames modes increase substantially the encoder/decoder complexity and computational requirements as compared to the H.263 baseline coder. The complexity of the improved *PB*-frames mode is slightly larger than that of the *PB*-frames mode. Besides the bidirectional prediction operation employed in the *PB*-frames mode, the improved *PB*-frames mode normally involves backward prediction (although trivial as described above) and forward prediction. The computational load associated with the improved *PB*-frames mode is also usually larger than that of the *PB*-frames mode. Although its backward prediction is trivial, and its bidirectional prediction does not involve a restricted motion search to obtain a delta motion vector, forward prediction of the improved *PB*-frames mode is usually very computationally intensive; hence, the computational advantage for the *PB*-frames mode.

Like the H.263 *PB*-frames mode, the H.263+ improved *PB*-frames mode requires more memory both at the encoder and the decoder as compared to the H.263 baseline coder. Since *P* and *B* pictures are processed at the same time, two pictures usually have to be stored simultaneously in memory. There is also a delay of one frame associated with both of the above modes, caused by encoding two instead of one picture. This becomes a problem in real-time applications.

D. Temporal, SNR, and Spatial Scalability Mode

1) Temporal Scalability: Fig. 11 shows the average luminance PSNR versus bit rate for the two sequences *Foreman* and *Akiyo*. Each graph contains four curves, showing PSNR's for decoded sequences at base display rates of 5, 10, 15, and 30 *P*-picture fps (*P*-fps). Each of the corresponding video sequences is then extended to 30 fps via temporal scalability, i.e., 5, 2, and 1 *B* pictures are inserted between anchor picture pairs in the sequences with base display rates of 5, 10, and 15 *P*-fps, respectively. The same fixed quantization level is used in both *P* and *B* pictures.

For the same quantization level and temporal distance from a reference anchor picture, the compression performance of a *B* picture is at least as good as that of a *P* picture due to the bidirectional prediction alternatives. In all temporal scalability

[3]The current *P* macroblock is the *P* macroblock that has the same spatial location as the current *B* macroblock.

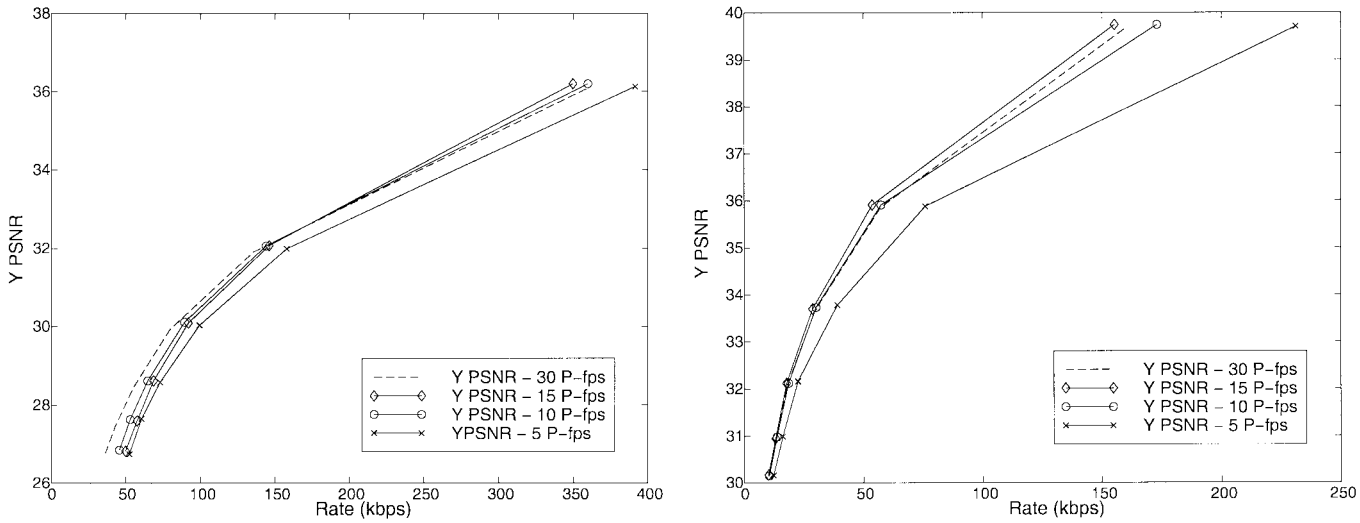


Fig. 11. Average luminance PSNR versus bit rate for (a) *Foreman* and (b) *Akiyo* when temporal scalability is used.

cases tested herein, *P* pictures require significantly more bits than *B* pictures. Thus, on a picture-by-picture basis, *P* pictures contribute more to increasing the total bit rate of a temporally scaled bit stream.

For a high-motion sequence like *Foreman*, the scaled 15 *B*-fps sequence requires a much lower bit rate than the scaled 5 *P*-fps sequence and a slightly lower bit rate than the scaled 10 *P*-fps sequence. Note that the scaled 15 *P*-fps sequence compares favorably to the reference 30 *P*-fps sequence. At low bit rates, the reference 30 *P*-fps sequence is about 0.5 dB better in terms of average PSNR than the scaled 15 *P*-fps sequence. This difference becomes negligible at bit rates above 100 kbits/s, and the scaled 15 *P*-fps sequence actually outperforms the reference 30 *P*-fps sequence at very high bit rates. In the case of *Akiyo*, the scaled 15 *P*-fps sequence again requires a much lower bit rate than the scaled 5 *P*-fps base sequence and a slightly lower bit rate than the scaled 10 *P*-fps sequence. Furthermore, the scaled 15 *P*-fps sequence achieves essentially the same compression performance level as that of the reference 30 *P*-fps sequence, and it is slightly better for bit rates above 30 kbits/s. Clearly, increasing the number of *B* pictures between consecutive *P* pictures results in decreased compression performance. However, use of a single *B* picture between anchor pictures yields comparable compression performance to a 30 *P*-fps sequence, with the added feature of a scalable display rate.

2) SNR Scalability: To evaluate SNR scalability, one- and two-layer, or nonscalable and scalable, bit streams are generated for the sequences *Akiyo* and *Foreman*. The quantization level is fixed for all pictures at the base layer of the scalable sequence. The enhancement layer quantization level is also fixed, being half that of the reference (base) layer. In our comparisons, the bit rate of the reconstructed enhancement layer sequence is defined as the total bit rate required for both the base and enhancement layers. The reconstructed enhancement layer sequence is then compared to a reconstructed sequence obtained using the nonscalable coder. The nonscalable coder employs the same fixed quantization level as that of the enhancement layer in the scalable coder.

Results for both sequences are shown in Fig. 12. Each graph consists of two curves, one representing the bit rate and luminance PSNR for the nonscalable coder, and the other representing the total bit rate and enhancement layer luminance PSNR for the scalable coder. The results show that a 1–2 dB drop in PSNR is sacrificed when generating a two-layer scalable bit stream. The multiple prediction options, including upward prediction from the base layer, forward prediction from the enhancement layer, and bidirectional prediction from both the base and enhancement layers, provide good inter frame prediction accuracy. However, the overhead associated with the additional layer and the DCT coefficient refinements offset the bit savings achieved by higher prediction accuracy.

3) Spatial Scalability: To evaluate spatial scalability, nonscalable and scalable bit streams are again generated for the sequences *Akiyo* and *Foreman*. The quantization level is fixed for all pictures at the base layer of the scalable sequence, which has a QCIF resolution. The enhancement layer sequence has a CIF resolution, and the quantization level is set to that of the reference (base) layer. Again, the bit rate of the reconstructed enhancement layer sequence is defined as the total bit rate required for both the base and enhancement layers. The reconstructed enhancement layer sequence is then compared to the reconstructed sequence obtained using the nonscalable coder. The nonscalable coder employs the same fixed quantization level as that of the enhancement layer in the scalable coder, and a CIF resolution video sequence.

Results for both sequences are shown in Fig. 13. Each graph consists of two curves, one representing the bit rate and luminance PSNR of the nonscalable coder, and the other representing the total bit rate and enhancement layer luminance PSNR of the scalable coder. The results for *Akiyo* show that a 1–2 dB drop in PSNR is sacrificed when generating a scalable bit stream. However, for the sequence *Foreman*, the scalable coder achieves as much as a 3 dB increase in PSNR over the

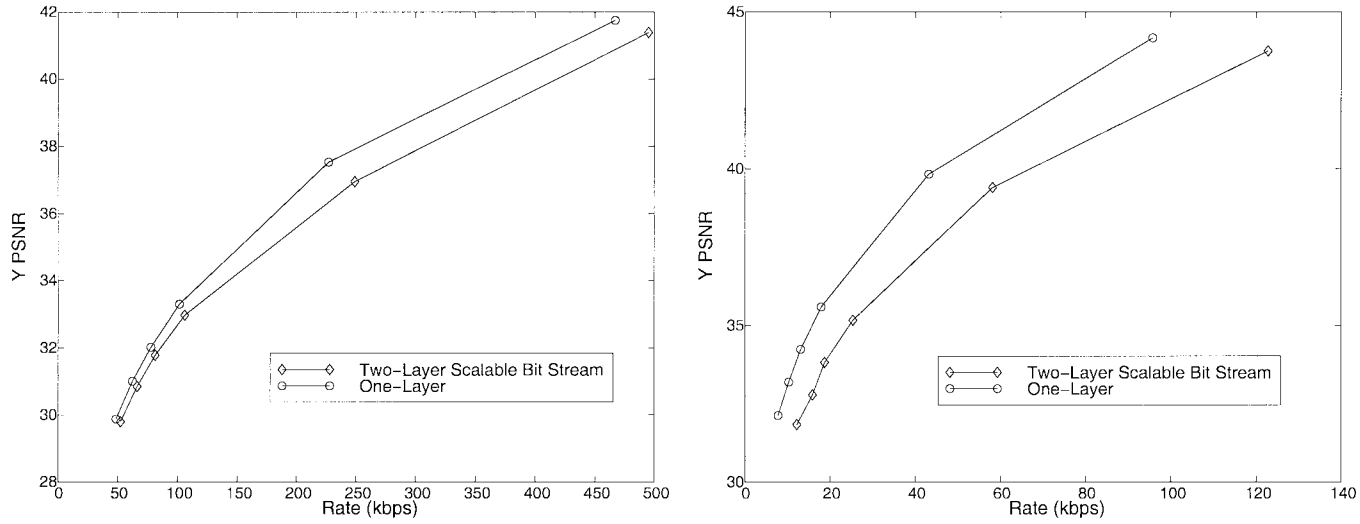


Fig. 12. Average luminance PSNR versus bit rate for (a) *Foreman* and (b) *Akiyo* when SNR scalability is used.

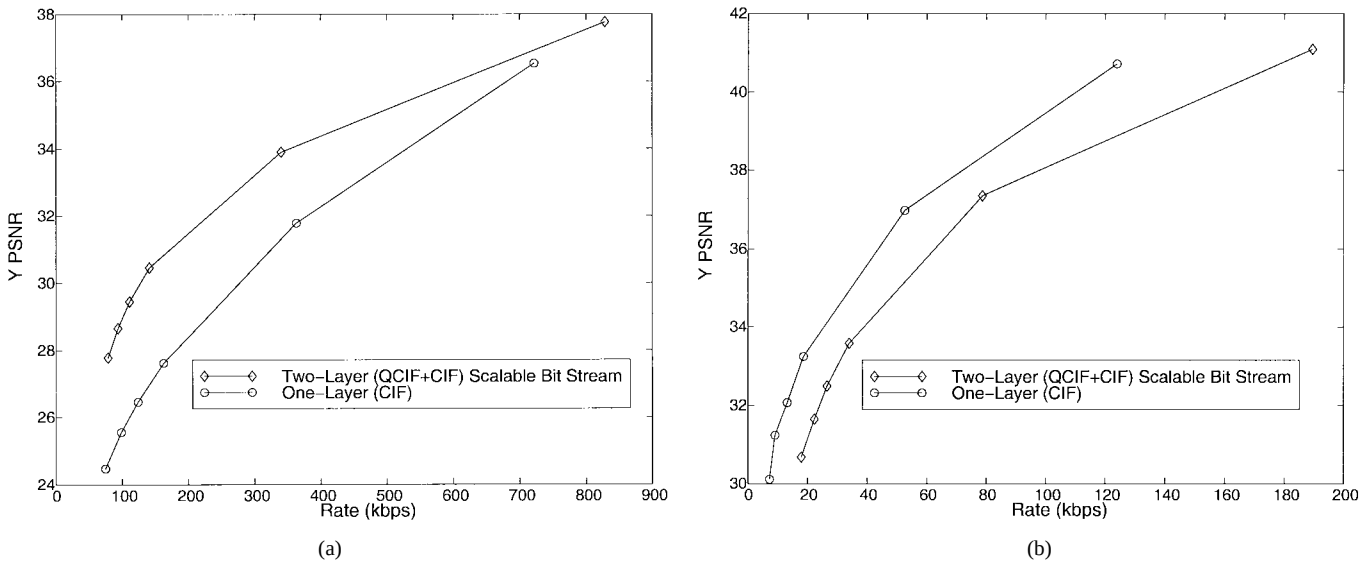


Fig. 13. Average luminance PSNR versus bit rate for (a) *Foreman* and (b) *Akiyo* when spatial scalability is used.

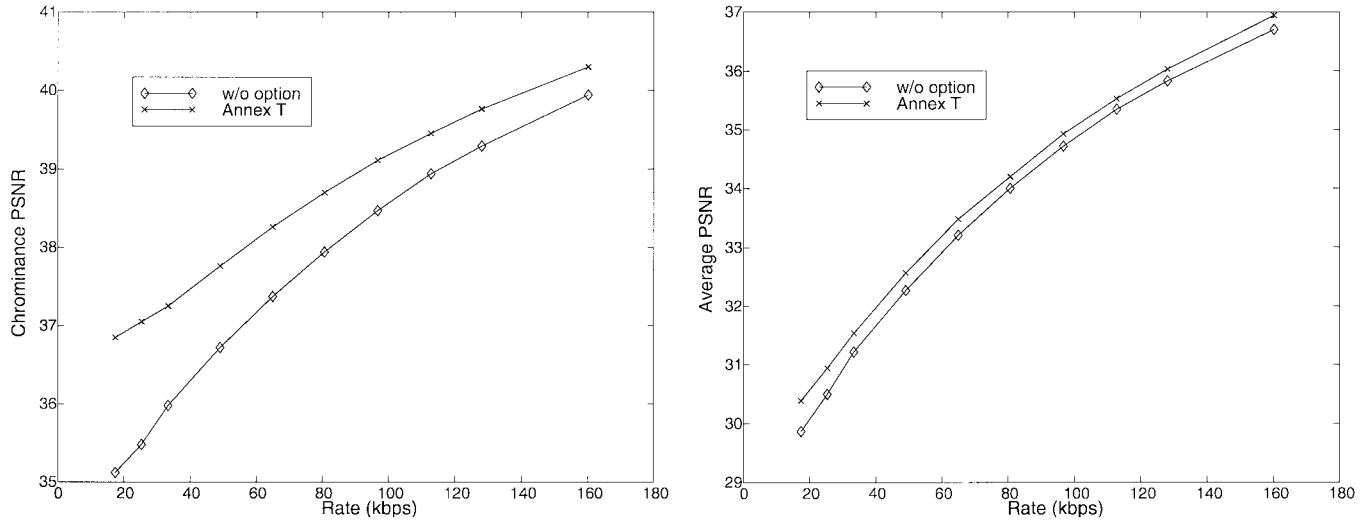
nonscalable coder. As *Foreman* contains high motion, camera motion, and occlusions, a significant proportion of *P*-picture macroblocks (an average of around 10% per picture) are intra coded in the nonscalable coder. In the scalable coder, most of this intra coding is performed at the base layer. Blocks that are intra coded by the nonscalable coder are, in the enhancement layer pictures of the scalable coder, predicted from the upsampled base layer pictures.

E. Alternative Inter VLC Mode

This mode allows the intra macroblock quantized DCT coefficient VLC's of the advanced intra coding mode to be used for some inter coded blocks. This mode of operation is useful at high bit rates, when short runs of zeros and large coefficients values are present, as the advanced intra coding mode run-length VLC's are designed for such statistics. Best results for this mode are obtained when fine quantizers are used, as can be seen in Table III. At very high bit rates, bit savings of as much as 10% can be achieved. Recall that the alternate inter VLC mode requires an extra scan of the quantized DCT coefficients to determine if the use of the intra table will be detectable by the decoder. The additional scan is also required at the decoder. However, this added complexity is negligible, especially in software applications. In fact, less than 2% additional encoding/decoding time is usually required.

F. Modified Quantization Mode

To fully illustrate the capabilities of this mode, the TMN-8 [14] rate control method is used for all simulations in this section. Fig. 14(a) shows the chrominance PSNR performance of the video sequence *Foreman* with and without the modified quantization mode enabled. From this figure, it is clear that



(a) (b)

Fig. 14. (a) Chrominance and (b) average PSNR performance for *Foreman* when the modified quantization mode is enabled.

TABLE III
AVERAGE BIT SAVINGS ON INTER FRAMES FOR THE ALTERNATE INTER VLC MODE

Sequence	Quantizer Step Size	Y PSNR	Bits (No option)	Bits (Annex S)	Bit savings
AKIYO	4	43.79	9354	8891	463 (5%)
	8	39.47	4128	4073	55 (1%)
	12	36.94	2411	2405	6 (0%)
FOREMAN	4	41.41	47659	44118	3541 (7%)
	8	37.15	22036	21175	861 (4%)
	12	34.73	13498	13201	297 (2%)
	16	33.12	9434	9320	114 (1%)
NEWS	4	42.48	21591	19562	2029 (9%)
	8	37.92	10706	10073	633 (6%)
	12	35.10	6750	6485	265 (4%)
	16	33.34	4764	4639	125 (4%)

the chrominance PSNR increases substantially at low bit rates. Naturally, this causes a drop in luminance PSNR as fewer bits remain to represent the luminance coefficients. However, this drop is rather insignificant, and the overall PSNR performance is usually improved. Fig. 14(b) shows that the overall PSNR performance is indeed higher when the modified quantization mode is enabled.

The modified quantization mode adds very little complexity to the coder. It requires only that syntax be added to represent the extended coefficient range and the modified difference quantizer. Using this mode, the resulting encoding speed is very close to that of the H.263 baseline coder.

G. Test Model Rate Control Methods: Evaluation

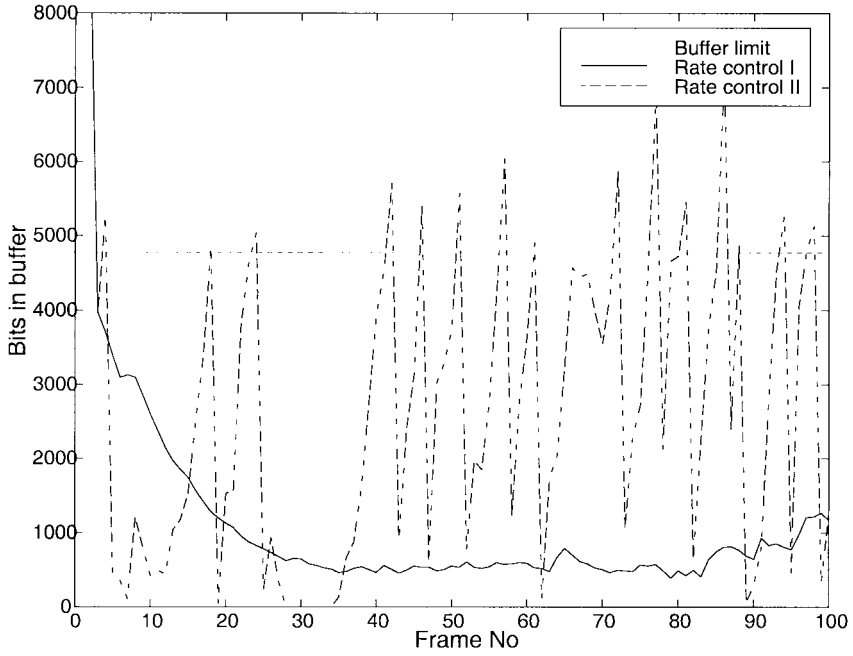
Both test model (TMN) bit rate control algorithms were implemented. In general, for a given bit rate, the two methods usually achieve similar PSNR level. However, the new rate control method introduced in TMN-8 [14], [15] achieves the target bit rate more accurately. Moreover, it keeps a buffer content well below the maximum level, thus reducing frame skipping as well as delay. If it is assumed that the decoder simply repeats the previous frame to replace a skipped frame, then the new bit rate control method performs better, in terms of PSNR, for a given bit rate. (It has also been reported in [15] that the new TMN-8 rate control method can outperform the alternate method by as much as 1.2 dB when the same number of frames are coded.)

Fig. 15 illustrates buffer fullness per frame for the video sequences *Foreman* and *Mother and Daughter* at 48 kbits/s and 10 fps. In this figure, rate control I is the new test model method, and rate control II is the alternate method (also known as TMN-5, TMN-6, or TMN-7 rate control method). Whenever buffer content reaches the model limit, frames are repeatedly skipped at the encoder until the buffer content is below the limit. In the case of rate control II, many frames are skipped, reducing temporal resolution, which can be critical in applications such as lip reading or sign language. Furthermore, the buffer content varies substantially from frame to frame (i.e., exhibiting high variance), introducing variable delays at the decoder. Finally, as buffer underflow occurs quite frequently, the available bandwidth is often not fully utilized. On the other hand, rate control I maintains a desirable, constant buffer fullness, offering a low and, more importantly, near-constant delay. Furthermore, the available bandwidth is fully utilized by avoiding buffer underflow.

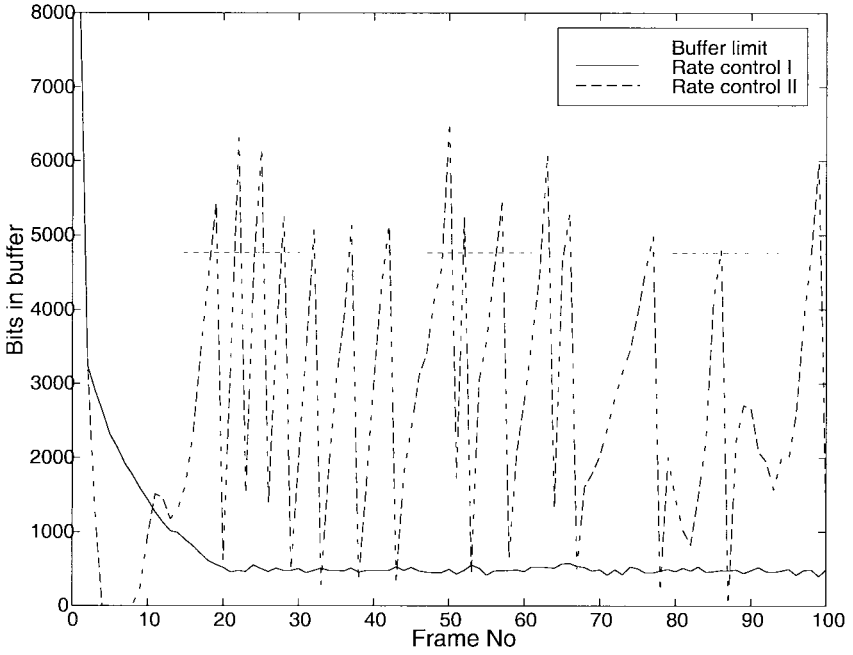
While this new rate control method is superior in terms of buffer fullness control performance, the number of computations increases, due mainly to variance calculations. In our implementation, this increases encoding time by approximately 5%. However, better computation–performance tradeoffs may be obtained by using, instead of variances, sum of absolute differences. Moreover, the calculation of variances may be incorporated in the motion estimation process, further reducing computational complexity.

H. Summary: Encoding/Decoding Speed and Compression Improvements of Individual Modes

Fig. 16 illustrates the added encoding computation times of individual H.263 and H.263+ optional modes tested above (except Annex O). The results were obtained by encoding 300



(a)



(b)

Fig. 15. Comparison of test model rate control methods based on encoder buffer regulation for (a) *Foreman* and (b) *Mother and Daughter* at 48 kbits/s and 10 fps.

frames of the video sequence *Foreman* at 64 kbits/s using our software coder [16] on a Pentium 200 MHz. The TMN-8 new rate control method was used. Decoding of an H.263 compliant bit stream can be easily performed in real time on the same Pentium PC. Additional computational resources required by the H.263+ modes are also negligible in our software decoder implementation, and real-time decoding of an H.263+ compliant bit stream can still be supported. The encoder's speed of the H.263 baseline coder with any H.263 or H.263+ individual mode enabled is at most 15% larger than that of H.263. Setting the *PB*-frames mode on results in a reduction in encoding time as only a restricted motion estimation operation is performed for the *B* picture of a *PB* frame. Encoding time is at most half of that of full search motion estimation when the fast search method is used (except for the H.263 *PB*-frames mode). Interestingly, the additional percentages of encoding times are similar for both the full- and fast-search motion estimation cases.

A summary of compression improvements resulting from the use of individual modes tested above (except Annex

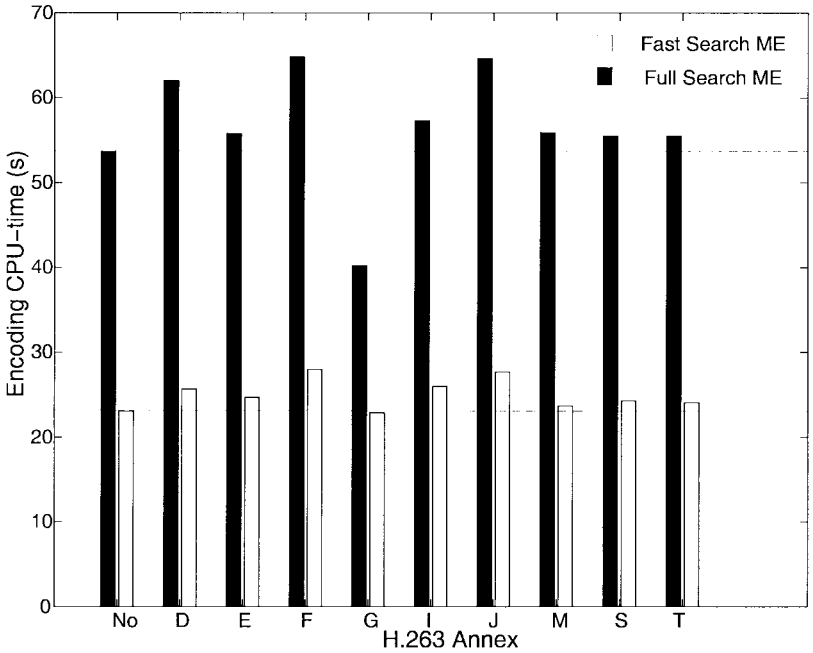


Fig. 16. Encoding CPU times for the H.263 and H.263+ modes for the video sequence *Foreman* at 64 kbits/s.

O), with the new TMN-8 rate control method, are given in Table IV. Results are presented for low and high bit rates using three QCIF video sequences at 10 fps: an active video sequence, *Foreman*, a sign language video sequence, *Silent*, and a typical head-and-shoulder video sequence, *Akiyo*. It can be observed that a given mode is not always suitable for any bit rate and/or any sequence. For example, the alternate inter VLC mode achieves compression gains only at high bit rates. Moreover, the deblocking filter mode may yield a decrease in PSNR, but the resulting picture subjective quality is usually much better. However, the latter mode may result in excessive blurriness at very low bit rates. Another observation is that the modified quantization mode does not lead to compression gains at high bit rates for low motion sequences, as the extended quantized coefficients range and the finer chrominance quantization are rarely used. The unrestricted motion vector mode shows PSNR improvements for sequences with motion across picture boundaries (in *Foreman*, for example), or at CIF and larger resolutions.

For high motion sequences (e.g., *Foreman*), the use of four motion vectors per macroblock and the use of an extended motion vector range improves compression performance significantly. Larger macroblock displacements present in high-motion sequences may not be accurately represented using the motion vector range specified by baseline H.263. Furthermore, the use of four motion vectors allows for better prediction and more accurate motion compensation. Therefore, for such high motion video sequences, the advanced prediction mode and/or the deblocking filter mode (both using four motion vectors per macroblock) and the unrestricted motion vectors mode should be employed. The use of four motion vectors can also be beneficial for low-motion video sequences. However, since many macroblocks are not coded, the overall compression gain is not as significant, and the use of four motion vectors may not justify the additional computational requirements.

TABLE IV
SUMMARY OF IMPROVEMENT IN PSNR (dB) FOR H.263 AND H.263+ INDIVIDUAL MODES AT (a) LOW BIT RATES AND (b) HIGH BIT RATES

Annex		FOREMAN 32 kbps	SILENT 24 kbps	AKIYO 8 kbps
No Option		30.44 dB	32.74 dB	33.9 dB
D		+0.61	+0.01	-0.01
E		+0.08	+0.04	-0.12
F		+0.28	+0.06	+0.19
G	B-pictures	-0.57	-0.05	+0.51
	P-frames	+0.11	+0.5	+0.6
I		+0.04	+0.11	+0.14
J		+0.18	+0.12	-0.24
M	B-frames	-0.17	+0.19	+0.54
	P-frames	+0.36	+0.62	+0.61
S		+0.02	+0.02	-0.02
T		+0.4	+0.2	+0.4

(a)

Annex		FOREMAN 128 kbps	SILENT 96 kbps	AKIYO 32 kbps
No Option		35.83 dB	38.84 dB	39.26 dB
D		+0.64	+0.01	-0.04
E		+0.06	+0.13	+0.08
F		+0.58	+0.13	+0.33
G	B-pictures	-1.64	-0.69	+0.61
	P-pictures	+0.37	+0.41	+1.05
I		0	+0.07	-0.03
J		+0.48	-0.02	-0.23
M	B-pictures	-1.21	-0.27	+0.74
	P-pictures	+0.62	+0.66	+1.12
S		+0.06	+0.23	+0.01
T		+0.2	+0.03	-0.05

(b)

VI. MODE COMBINATIONS: COMPRESSION PERFORMANCE AND COMPLEXITY

With the large number of possible mode combinations, it becomes difficult for implementers to select combinations that

TABLE V
COMBINATIONS OF MODES: COMPRESSION PERFORMANCE AND ENCODING TIME FOR THE TYPICAL VIDEOPHONE SEQUENCE *AKIYO* AT (a) 8 kbps AND (b) 32 kbps BIT RATES AND FOR THE ACTIVE VIDEO SEQUENCE *FOREMAN* AT (c) 32 kbps AND (d) 128 kbps BIT RATES

Combined Annexes Level 1	FAST ME		FULL ME	
	PSNR P-pictures	Encoding time	PSNR P-pictures	Encoding time
Baseline	33.9 dB	16.9 sec	33.9 dB	41 sec
I-J-T	+0.74	+28%	+0.74	+25%

(a)

Combined Annexes Level 1	FAST ME		FULL ME	
	PSNR P-pictures	Encoding time	PSNR P-pictures	Encoding time
Baseline	39.26 dB	18.8 sec	39.26 dB	35.4 sec
I-J-T	+0.38	+28%	+0.44	+25%

(b)

Combined Annexes Level 3	FAST ME			FULL ME		
	PSNR		Encoding time	PSNR		Encoding time
	P-pictures	B-pictures		P-pictures	B-pictures	
Baseline	30.44 dB	N/A	20.8 sec	30.44 dB	N/A	56.4 sec
(I-J-T)+(D)+(F-S)	+0.79	N/A	+39%	+0.55	N/A	+26%
(I-J-T)+(D)+(F-S-M)	+1.24	+0.56	+27%	+0.88	+0.3	+22%

(c)

Combined Annexes Level 3	FAST ME			FULL ME		
	PSNR		Encoding time	PSNR		Encoding time
	P-pictures	B-pictures		P-pictures	B-pictures	
Baseline	35.83 dB	N/A	25.3 sec dB	N/A	35.83 dB	54.9 sec
(I-J-T)+(D)+(F-S)	+1.05	N/A	+34%	+1.1	N/A	+21%
(I-J-T)+(D)+(F-S-M)	+1.52	-0.42	+15%	+1.53	-0.27	+19%

(d)

are suitable for their applications. The ITU-T video experts group decided to include nonnormative mode combinations as guidelines for implementers. Appendix II of ITU-T Recommendation H.263 Version 2, entitled "Recommended Optional Enhancement," describes levels consisting of recommended mode combinations that are obtained based on the performance of individual modes. The performance criteria are the improvement in subjective quality, the impact on delay, and the additional complexity, computation, and memory demands.

The Level 1 preferred combination of modes includes the advanced intra coding (Annex I), deblocking filter (Annex J), supplemental enhancement information, full-frame freeze only (Annex L.4), and modified quantization (Annex T) modes. The Level 2 preferred combination of modes includes, in addition to the Level 1 modes, the unrestricted motion vector (Annex D), slice structure (Annex K), and reference picture resampling and the implicit factor of four only (Annex P) modes. Finally, the Level 3 preferred combination of modes includes Level 2 and Level 1 preferred modes, and the advanced prediction (Annex F), improved PB frames (Annex M), independent segment decoding (Annex R), and alternate inter VLC (Annex S) modes.

In our experiments, an error-free environment is assumed. Thus, the H.263+ modes for error resilience (Annexes K, N, and R) are here excluded. Similarly, the full-frame freeze option is not included in our simulations as it provides enhanced display capabilities, but does not impact performance. The implicit factor of four resampling option is currently not available in [16].

Table V presents results for the Level 1 and Level 3 mode combinations for the video sequences *Akiyo* and *Foreman*, respectively. Based on our experiments, using a higher level of mode combinations provides better compression performance, especially for highly active video sequences such as *Foreman*, by as much as 1.5 dB at high bit rates. Moreover, the encoding time of an H.263+ encoder employing a combination of modes that includes the improved PB-frames mode is reduced since many computationally expensive operations are not required by the B picture of a PB frame. Finally, note that the encoding time is, at most, approximately 25% greater, even for the Level 3 combinations of modes.

REFERENCES

[1] K.-H. Tzou, H. G. Musmann, and K. Aizawa, *IEEE Trans. Circuits Syst. Video Technol.* (*Special Issue on Very Low Bit Rate Video Coding*), vol. 4, pp. 213–367, June 1994.

[2] W. Li, Y.-Q. Zhang, and M. L. Liou, *Proc. IEEE (Special Issue on Advances in Image and Video Compression*), vol. 83, pp. 135–340, Feb. 1995.

[3] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: A review," *IEEE Trans. Image Processing*, vol. 3, pp. 568–609, Sept. 1994.

[4] B. Girod, K. B. Younes, R. Bernstein, P. Eisert, N. Farber, F. Hartung, U. Horn, E. Steinbach, T. Wiegand, and K. Stuhlmuller, "Recent advances in video compression," in *IEEE Int. Symp. Circuits Syst.*, Feb. 1996.

[5] B. Girod, "Advances in digital image communication," in *Proc. 2nd Erlangen Symp.*, Erlangen, Germany, Apr. 1997.

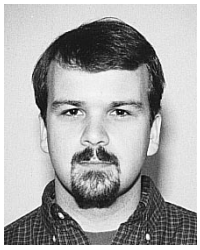
[6] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *ITU-T Recommendation H.263*, Mar. 1996.
[7] ITU Telecom. Standardization Sector of ITU, "Video coding for low bitrate communication," *Draft ITU-T Recommendation H.263 Version 2*, Sept. 1997.
[8] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture.* Boston, MA: Kluwer Academic, 1995.
[9] K. P. Rao and P. Yip, *Discrete Cosine Transforms: Algorithms, Advantages, Applications.* New York: Academic, 1990.
[10] J. Johnston, N. Jayant, and R. Safranek, "Signal compression based on models of human perception," *Proc. IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.
[11] B. Girod, E. Steinbach, and N. Faerber, "Comparison of the H.263 and H.261 video compression standards," in *Standards and Common Interfaces for Video Information Systems*, K. R. Rao, Ed., *Critical Reviews of Optical Science and Technology*, Philadelphia, PA, Oct. 1995, vol. 60, pp. 233–251.
[12] N. Faerber, B. Girod, and E. Steinbach, "Performance of the H.263 video compression standard," *J. VLSI Signal Processing: Syst. for Signal, Image, and Video Technol. (Special Issue on Recent Development in Video: Algorithms, Implementation and Applications*), no. 17, pp. 101–111, 1997.
[13] J. Wen and J. D. Villasenor, "A class of reversible variable length codes for robust image and video coding," in *Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997.
[14] ITU Telecom. Standardization Sector of ITU, "Video codec test model near-term, Version 8 (TMN8), Release 0," H.268 Ad Hoc Group, June 1997.
[15] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay vcommunications," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[16] Signal Processing and Multimedia Lab., Univ. British Columbia, "TMN 8 (H.263+) encoder/decoder, Version 3.1.3," TMN 8 (H.263+) codec, Feb. 1998.
[17] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, "Error resilience support in H.263+," *IEEE Trans. Circuits Syst. Video Technol.*, this issue. pp. 867–877.
[18] M. Gallant, G. Cote, and F. Kossentini, "A computation constrained block-based motion estimation algorithm for low bit rate video coding," *IEEE Trans. Image Processing*, submitted Mar. 1997, revised Mar. 1998.

Guy Côté (S'93) received the B.S. degree in electrical engineering from the Royal Military College of Canada, Kingston, Canada, in 1993.

From 1993 to 1996, he served as a Communications Officer in the Canadian Army. He is currently a Ph.D. student at the Signal Processing and Multimedia Laboratory, Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada. His research interests include image and video compression and coding, error resilience for video communications, and joint source–channel coding.

Berna Erol received the B.S. degree in computer and control engineering from the Istanbul Technical University, Turkey, in 1994, and the Master of Applied Science degree from the Department of Electrical and Computer Engineering, University of British Columbia, in 1998.

After completing her undergraduate studies, she worked as a Scientific Engineer at the University of British Columbia, Canada, for two years where she developed software applications for DSP's. Currently, she is a Ph.D. student at the same University. Her research interests include software implementation of video encoders/decoders and object-based video coding.



Michael Gallant (S'97) received the B.A.Sc. degree magna cum laude in electrical engineering from the University of Ottawa, Ottawa, Canada in 1995.

From 1995 to 1996, he worked as an Engineer for Nortel. He is currently a Ph.D. student at the Signal Processing and Multimedia Laboratory, Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada. His research interests include image and video compression, video communications, and signal processing.

Faouzi Kossentini (M'89), for a photograph and biography, see this issue, p. 848.