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

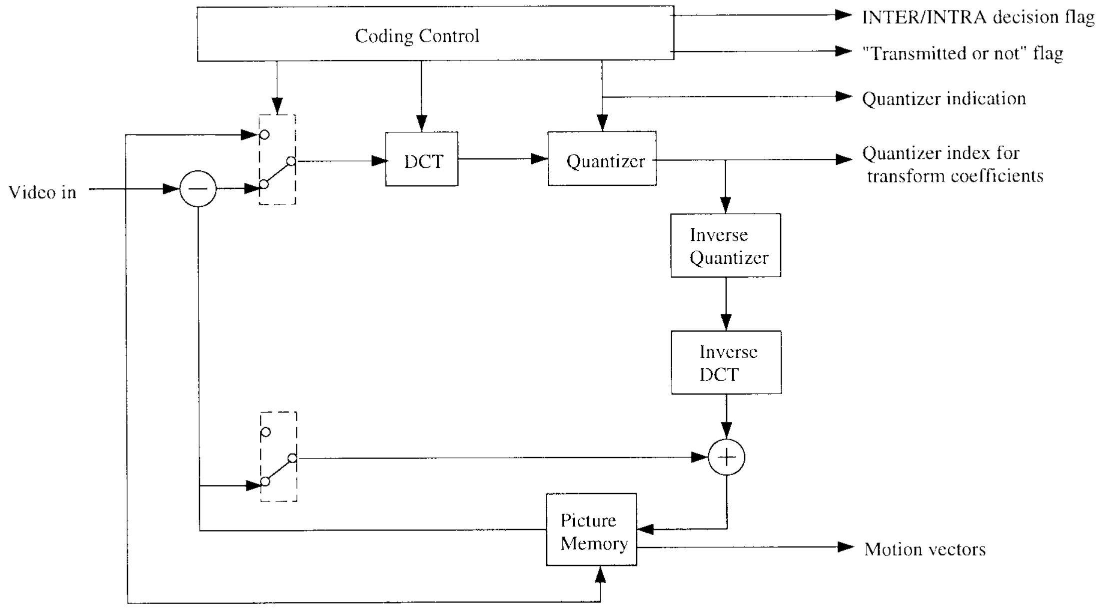

Fig. 1. H.263 video encoder block diagram.

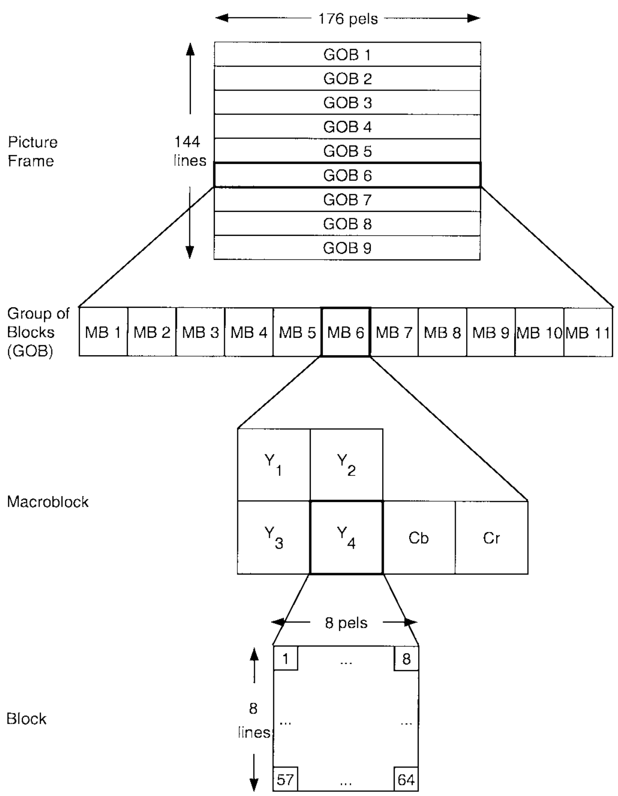

Fig. 2. H.263 picture structure at QCIF resolution.

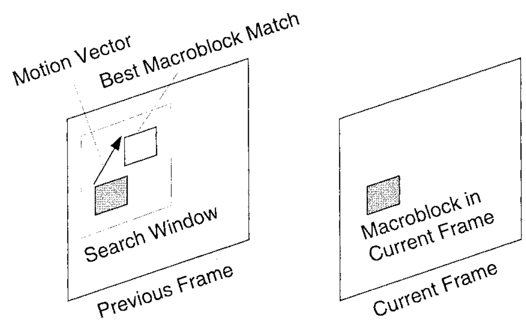

Fig. 3. H.263 source coding algorithm: motion compensation.

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 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

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

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.

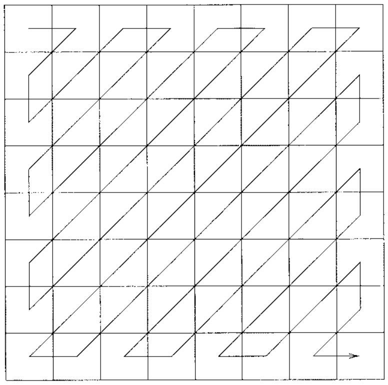

Fig. 4. Zigzag scan pattern to reorder DCT coefficients from low to high frequencies.

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.

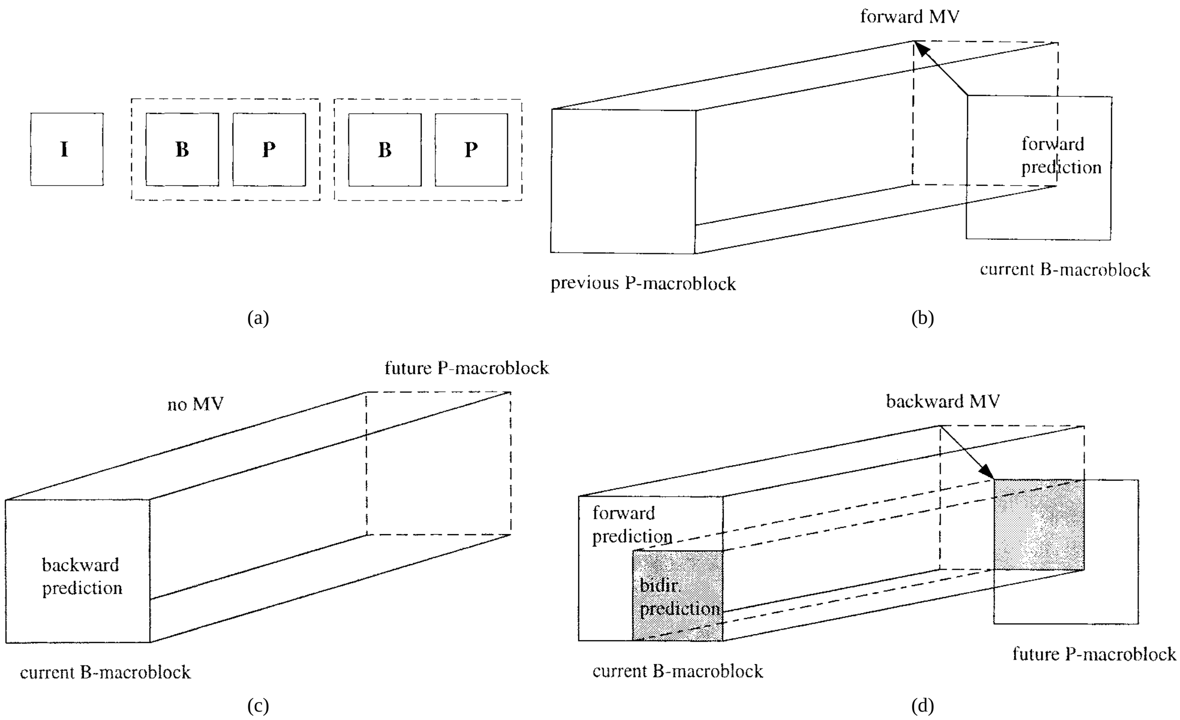

Fig. 5. Improved *PB* frames. (a) Structure. (b) Forward prediction. (c) Backward prediction. (d) Bidirectional prediction.

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.

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

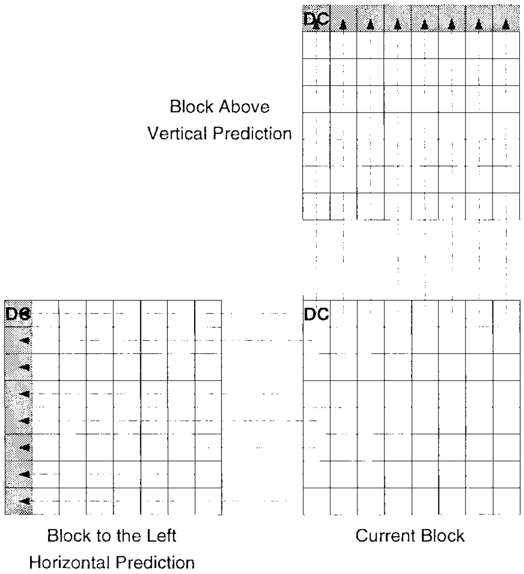

Fig. 6. Neighboring blocks used for intra prediction in the advanced intra coding mode.

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

[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++.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.