# Data Partitioning and Reversible Variable Length Codes for Robust Video Communications

**Adam H. Li[1], Surin Kittitornkun[2], Yu-Hen Hu[2],
Dong-Seek Park[3], and John Villasenor[1]**

[1] Electrical Engineering Department, University of California at Los Angeles, 405
Hilgard Avenue, Los Angeles, CA 90024-1594
[2] Electrical and Computer Engineering Department, University of Wisconsin at Madison,
1415 Engineering Drive, Madison, WI 53706-1691
[3] Samsung Electronics Company, Suwon, South Korea

### *Abstract*

Low bit-rate multimedia communication over wireless channels has received much attention recently. A key challenge in low bit-rate wireless communication is the very high error rate during transmission. This demands error resilient services that exhibit graceful performance degradation while operating in a highly noisy channel. Among video, audio and other multimedia communication modalities, we focus on the error-resilient transmission of video signals over wireless channels in this paper. Specifically, efforts in developing H.263++ Annex V error resilient Data Partitioning with Reversible Variable Length Code (RVLC) are described in detail. The performance over error-prone channels is analyzed and the effectiveness of the syntax is demonstrated with extensive simulations.

## I. Introduction

Video compression and transmission over traditional wired-channel communication systems are gaining wider acceptance. Video conferencing is regularly held to conduct business and for other occasions. Internet-based video-phone services such as

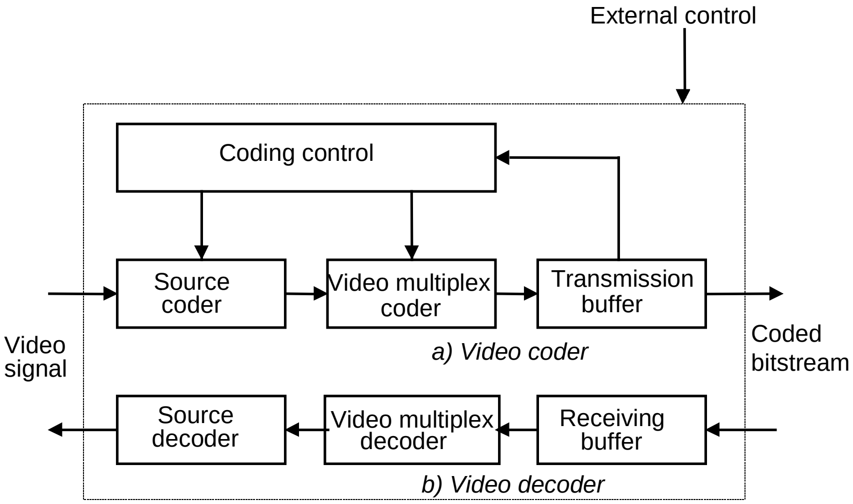
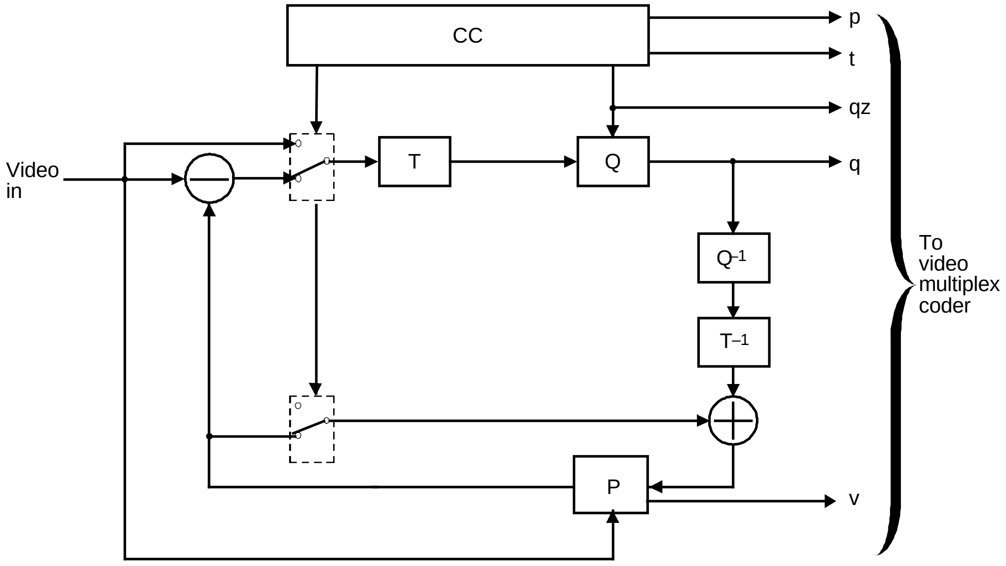


**Figure 1. H.263 Video codec system block diagram [1].**

Netmeeting have also been utilized by many users. A driving force behind this phenomenal growth of video communication services is the establishment of international video codec standards such as ITU H.261 and H.263.

H.263 is developed by International Telecommunication Union – Telecommunication Standard Sector (ITU-T) as a standard for compressing the moving picture component of audio-visual services at low bit rates. It was approved in 1996, and was the best standard for practical video telecommunication. Even though it was originally designed for a target bit rates of under 64 Kbit/s, it works well for higher bit rates too. The application target bit rate range of H.263 has subsequently been broadened to 10-2048 kbit/s. In 1998, the second version of H.263, also known as H.263+, was approved. It added a number of very useful new optional features that provide improvements in coding efficiency, flexible video format, scalability, etc. In addition, the effective bit rate range was further extended. Built on these prior successful experiences, the ITU-T Advanced Video Coding Experts Group (SG16/Q15) is now working on the second revision of H.263, code name H.263++, which is expected to be finalized by the year of 2000. The basic block diagram of a H.263 codec (coder/decoder) is shown in Figure 1 [1].

The video input first is encoded into a bitstream by the source coder. Then it is multiplexed with the audio portion of the signal and sent to the channel through the transmission buffer. The details of the source coder block is illustrated in Figure 2 [1]:



| | |
|---|---|
| T | Transform |
| Q | Quantizer |
| P | Picture Memory with motion compensated variable delay |

| | |
|---|---|
| CC | Coding control |
| p | Flag for INTRA/INTER |
| t | Flag for transmitted or not |
| qz | Quantizer indication |
| q | Quantizing index for transform coefficients |
| v | Motion vector |

**Figure 2. Block diagram of H.263 source coder [1].**

During the source coding, the video frames are encoded with either an intra-frame coding mode where the content of the entire frame is encoded, or an inter-frame coding mode where the content of the current frame is to be predicted from reference frames via motion compensation. Discrete cosine transform (DCT) is applied to each frame in the intra-frame coding mode, and to the residue after motion compensation in the inter-frame coding mode. Scalar quantization is applied to the DCT coefficients to yield binary integers. These binary integers are then encoded with variable length coding to form a single video bitstream.

One of the major goals of H.263++ is to extend the wire/line based video communication services to a mobile communication environment. One major challenge is that the bit error rates of the mobile wireless channels are usually orders of magnitude higher than those of the wired channels. For a video compression scheme to work under those conditions, one will need to address the issue of dealing with this error prone environment.

Error Resilience is identified as one of the Key Technical Areas for the H.263++ effort. Annex V is the latest addition to H.263++ in this area. To improve the performance of the H.263+ standard under error prone environment, a data-partitioning structure based on reversible variable length codes (RVLC's) was proposed by University of California at Los Angeles (UCLA) and Samsung Electronics [2,3]. A model implementation of this structure was made [4], and qualitative measurement of the performance of the data-partitioning structure was carried out [5-9]. The proposal has been adopted as Annex V of the H.263++ standard during the video expert meeting of SG 16/Q 15 at Monterey, California in February 1999 [10], and the Annex V text was finalized in the recent meeting at Red Bank, New Jersey in October 1999 [11].

# II. Error Resilience Measures in H.263++ Annex V

Error resilience can be accomplished via data partitioning and unequal error protection. Data partitioning involves segmenting the bit stream into independent segments so that the decoding of each segment will not be affected by errors occurring in other segments. Unequal error protection imposes a more reliable (and more costly) coding scheme to important bits and leaves less important bits unprotected.

## II.A Impact of Transmission Error on Variable Length Code

Variable length coding (VLC) is a special case of entropy coding. Since there are no delimiters used in an entropy coder to separate the successive binary code words such codes are highly susceptible to transmission errors since even a single bit error may cause significant changes in the decoded symbol stream. Some measure of protection can be obtained by wrapping the VLC in an error correcting code, but this introduction of additional redundancy defeats the original purpose of bit reduction using the entropy code.

In the baseline H.263+ syntax coded block case, when an error burst hits the middle of the block, the decoder loses its synchronization, and the rest of the data in that block becomes useless. This is a tremendous waste of resources in terms of encoding/decoding time and channel usage. It would be much more desirable to be able to salvage some information from the VLC coded sequence even in the presence of bit error.

## II.B Reversible Variable Length Coding (RVLC)

In the Annex V of H.263++, a variant of Huffman code, called *Reversible Variable Length Code* (RVLC) has been proposed [12]. It was shown that RVLC will offer essentially the same coding efficiency as all the other Huffman code while imposing a symmetry structure of each variable length code word. As such, each RVLC code can be decoded in both the forward and the backward directions. Therefore, if there are error bursts detected during RVLC decoding, the decoder can jump to the end of the segment and start decoding backwards. As a result, only the part of the block that is actually hit by the error burst is lost, and more data is recovered in the RVLC case. This is illustrated in Figure 3 below. The ability of RVLC to recover data from a block with errors gives H.263++ more error resilience.

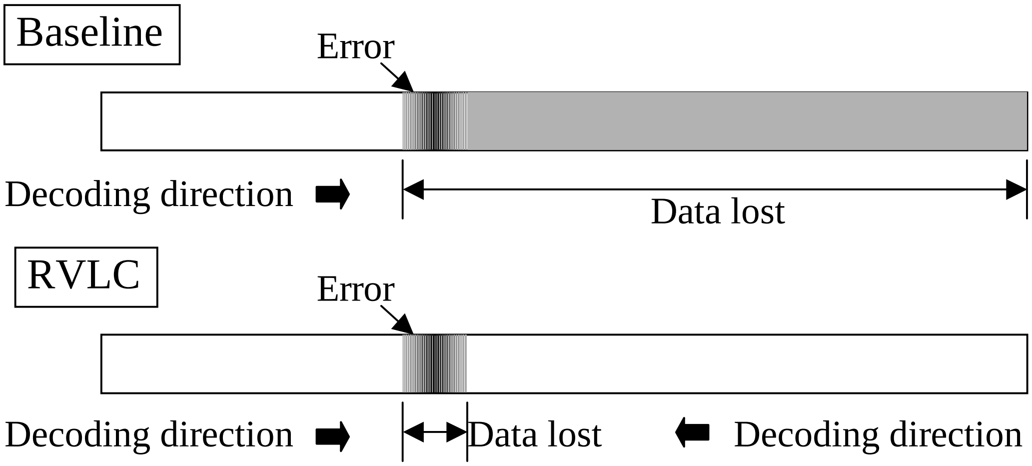


**Figure 3. Comparison between baseline H.263 VLC decoding and H.263++ RVLC decoding**

## II.C Syntax and Data Partitioning Modifications for RVLC

In order to take advantage of the reverse decoding capability of the RVLC, modification to the baseline H.263+ syntax is needed.

### *II.C.1 Bitstream Data Structure*

In baseline H.263+, a trade-off method adopted to deal with the error propagation problem is to designate a group of blocks (GOB) as unit for packetization. More specifically, in the H.263+ baseline syntax, each frame is divided into several Group-of-Block's (GOB). Within each GOB, the Macro-Block's (MB) that constitute that GOB are placed in sequence with the corresponding MB header, Motion Vector (MV) data and DCT coefficient data (see Figure 4). The GOB structure has improved the situation slightly by inserting re-sync markers at the beginning of each GOB; hence the errors do not propagate across GOB's. However, the propagation of errors is still a major problem for such syntax, because most of the fields of the MB are variable-length and in particular the DCT coefficient data, which is run-length, coded. A single bit error could render the rest of the data in that GOB unusable, which means that the decoder could only regain its synchronization at the next re-synchronization marker (cf. Figure 5). Also, since each GOB has fixed number of MB's, the interval between these inserted markers varies greatly depending on the content of the coded video sequences. Often, the use of GOB structure may involve too much overhead when the motion is low (where each MB can be as short as one bit), or may not provide enough protection when the motion is high (where the re-sync markers are still too far apart). As one can see, baseline H.263+ syntax does not offer enough error-resilience capability for transmission through error prone channels.

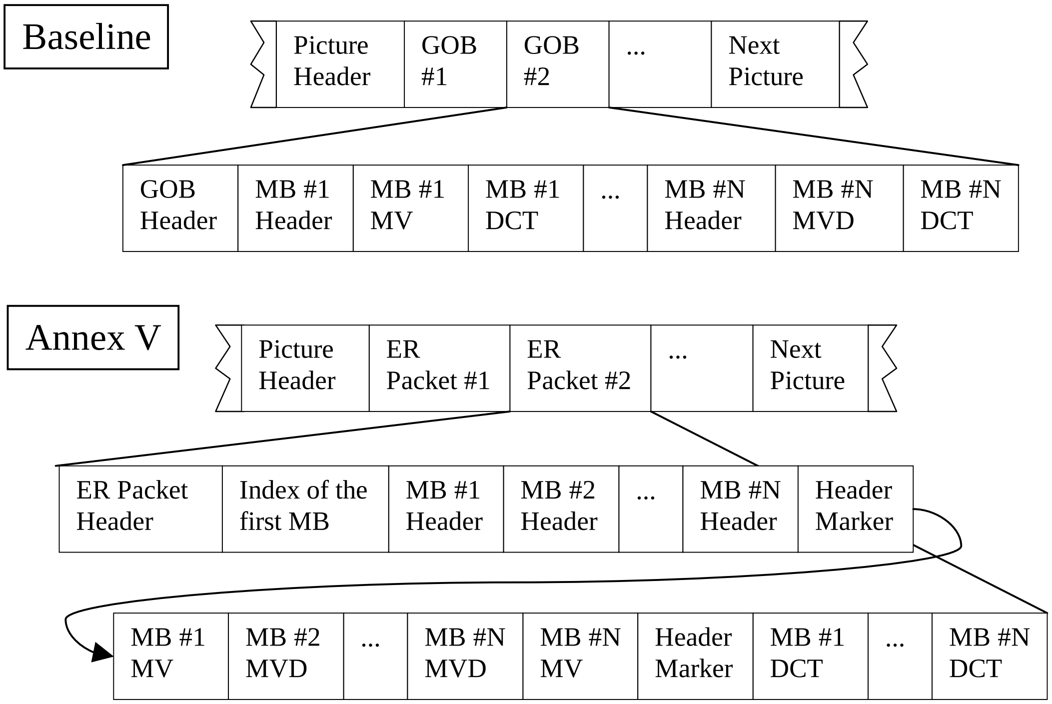


**Figure 4. Comparison of bit stream syntax in both H.263+ and Annex V of H.263++**

*II.C.2 Single Threaded Motion Vector Prediction*

To achieve a high compression ratio, differential motion vectors are used to code inter frames in H.263+. In baseline H.263+, for each coded MV, the mean value of three neighbor's motion vector is used as the predictor (see Figure 5). The difference between the absolute value of the MV of that block and the predictor is thus coded. However, this scheme is not suited for reverse decoding, because one can not get the correct predictor when decoding in the backward direction. Therefore, a new single threaded MV prediction scheme is proposed for Annex V [1]. As illustrated in Figure 4 above, the single threaded scheme MV prediction codes the MV's in a single one-dimensional thread. Thus the decoder can go fully backward when needed, which solves the problem of utilizing the reverse decoding in differential motion vector coding.

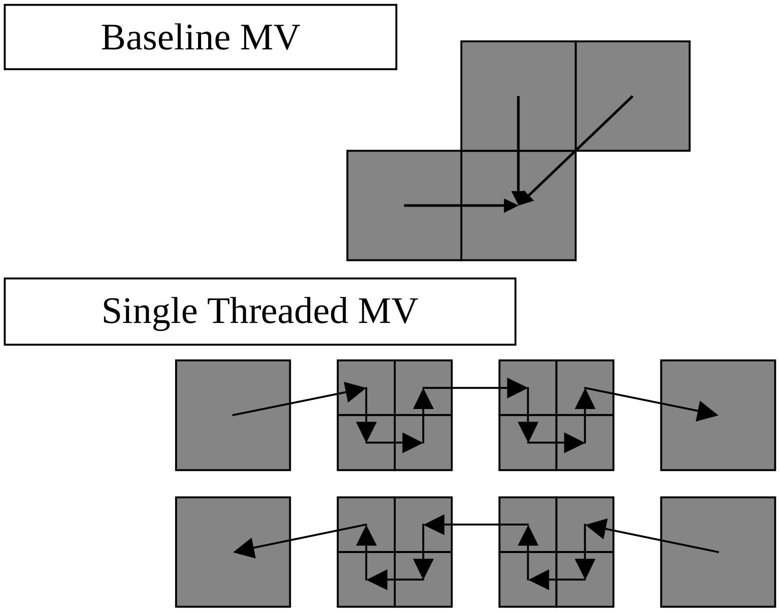


**Figure 5. Comparison between differential MV estimation and single-threaded MV**

*II.C.3 Data Partitioning Packet Structure*

The other modification needed to make use of RVLCs involves adding the data partitioning packet structure (see Figure 6). In baseline H.263+, the Header, MV and DCT data of each MB is ordered in the packet in an interleaved format. While the Header and MV data can be coded in RVLC, the DCT data is run-length coded, and is not backward decodable. In order to overcome this problem, all the data for all the MB's in the packet are collected and coded in three separate partitions, namely the header

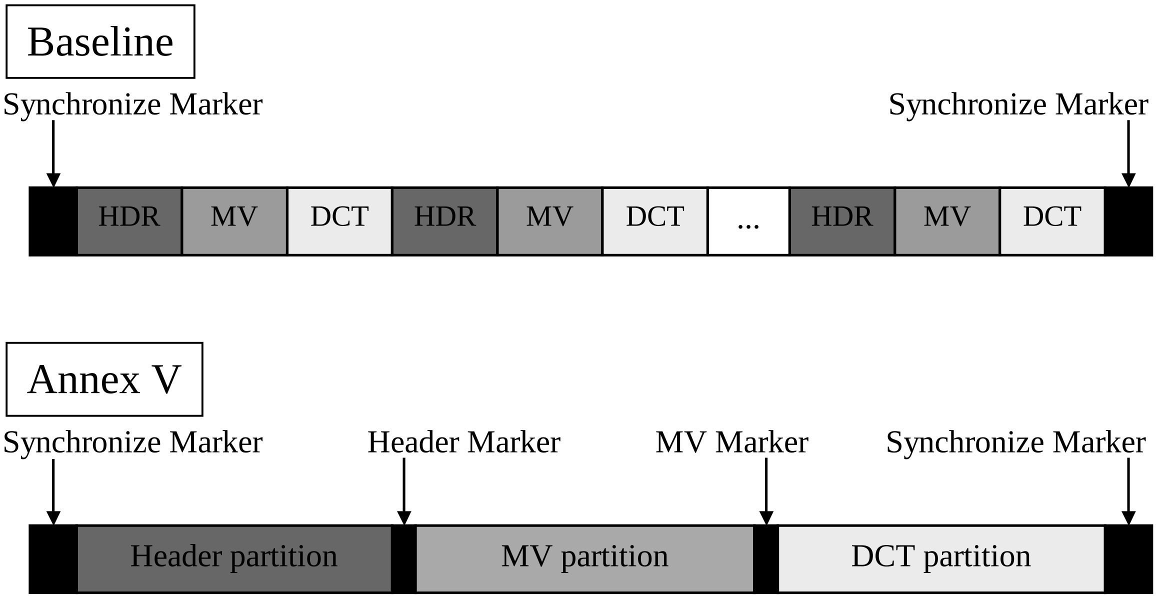


**Figure 6. Comparison of packet structure between baseline and Annex V syntax.**

partition, the MV partition and the DCT partition. While it is still impossible to perform reverse decoding on the DCT partition (since it is run-length coded), the other two partitions (Header and MV) will be protected by enabling reverse decoding on these RVLC coded partitions.

Annex V also utilizes a pseudo-fixed-length packaging scheme; the data of sequences are coded into packets of approximately the same length. This new addition helps to overcome the problem mentioned before for the baseline H.263+ that the inserted re-sync marker have irregular spacing depending on the coded video content. With the pseudo fixed-length packaging, the markers at the boundary of the packets are more evenly distributed to help the decoder to regain synchronization. Additionally, the separation of the markers (i.e. the packet size) can be selected to adapt to the error rate of the particular channel through which the sequence is sent.

In summary, to achieve better error resilience for H.263+, a new syntax of Annex V is adopted. It utilizes the error resilience capability of RVLC, and introduces new schemes of Single Thread MV Prediction, Data Partitioning Structure, and Pseudo Fixed-length Packaging. With all these new features, Annex V will greatly enhance the performance of H.263++ in an error-prone environment.

# III. Implementation and Simulation Results

Extensive simulations were performed based on the above ideas, and the results of these simulations were used in formulating Annex V. The simulation results show that the Annex V syntax does give significantly better objective and subjective results under almost all test conditions [4-8].

## III.A Simulation method and common conditions

The features of Annex V are implemented using the UBC (University of British Columbia) codec as a starting point for modifications. The Advanced Video Coding Expert Group (SG16/Q15) has developed a set of common testing conditions for simulation of video bitstream transmitted through mobile communication channels [13, 14]. The results presented in this paper conform to the simulation conditions outlined in those documents. During simulation, 4000 frames of each video sequence are encoded. Annex D, F, I, J and T are invoked, and an initial quantizer QP of 20 is set with the encoder. The encoded bit stream is multiplexed with AL-3 multiplexer which allocates 75% of transmission bit rate to video packets and 25% to other packets, which approximates the channel capacity allocation rate under normal conditions. Error patterns derived from a variety of real systems are then applied to the CRC checked transmitted binary digits.

On the decoder side, the multiplexer processes the data, and passes the bitstream to the decoder while discarding the CRC check results. The decoder (with the TCON error concealment option invoked, in order to verify that the new syntax can improve the performance where the error concealment can not) then decodes the bitstream accordingly, and PSNR is calculated according to specifications in previous documents pertaining to the simulation environment [13,14]. The specified method for PSNR calculation is carefully designed to ensure fair comparison when frame dropping happens.

Six sequences are used in the simulation. For simulation of 64 kbit/s channels, sequences Foreman, Hall, Container, News, Silent Voice and Glasgow Tour are coded with fps rate of 7.5 fps, 10 fps, 10 fps, 10 fps, 15 fps and 15 fps respectively. For simulation of 32 kbit/s channels, sequences Hall, Container, News and Silent Voice are coded with fps rate of 10 fps, 10 fps, 10 fps and 15 fps respectively. These video sequences are chosen to represent a spectrum of scenarios for the H.263++ standard from relatively static scenes with little motion to rapidly moving scenes with constant camera angle switch. The fps rates are likewise selected to cover a range of possible transmission channel capacities the codec needs to accommodate. Six WCDMA error patterns are applied in simulation for the 64 kbit/s bitstreams. As for the 32 kbit/s bitstreams, each of the 10 dB and 20 dB Rayleigh patterns for DECT systems with two different Doppler speeds, 1.4 and 14 km/h are applied.

## III.B Simulation results for Forced Update Rate of 132

The simulation results for an update rate of 132 are shown below for WCDMA and DECT error patterns. A rate of 132 means that each macroblock is intra coded at least once every 132 frames. This is a relatively slow update rate, meaning that the video coding will be efficient, but that any image artifacts due to errors can persist for long periods of time. By contrast, a fast update rate (such as the rate of 5 considered in the next section) provides greater error recovery capabilities but at the cost of lower coding efficiency due to the larger number of intra coded blocks. One reason for considering two update rates in this work is to ensure that the methods presented here can be evaluated for fast and slow update rates. Perhaps more importantly, this shows that the techniques proposed here give advantages over and above the robustness realized by simply speeding up the update rate in a noisy environment.

To illustrate the improvement of the image quality, the gain of PSNR for the Y component is averaged over all the video sequences that are tested under the particular error pattern.

| Error Pattern | WCDMA | | | | | | DECT | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 10 dB | | 20 dB | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 14 km/h | 1.4 km/h | 14 km/h | 1.4 km/h |
| Improvement of luminance PSNR (dB) | +3.55 | +3.79 | +3.51 | +1.80 | +1.95 | +1.35 | +3.66 | +1.03 | +3.46 | -0.13 |

Table 1. Improvement of PSNR – Y (Annex V vs. Baseline) under Forced Update Rate of 132 (averaged over all sequences).

## III.C Simulation results for Forced Update Rate of 5

In addition to the simulation above, in which all sequences are encoded with a Forced Updating Rate of 132 (the least frequent INTRA update required by the standard), same simulation is also done for sequences coded with Forced Update Rate of 5 (more frequent

INTRA update). The Forced Update Rate is a parameter for the encoder to specify how many times a MB can be predictively coded before it is forced to be coded in INTRA. This is originally included in the standard to prevent the accumulation of the transformation error from IDCT (Inverse DCT). The standard specifies a least frequent value of Forced INTRA Update of 132, based on the performance of currently used IDCT's. However, there is a concern that this feature will have some error resilience effect and may make the adoption of Annex V unnecessary. So simulation is done at both the Forced Update Rates of 132 and 5. Results show improvement under both conditions.

The simulation results for Forced Update Rate of 5 are shown in Table 2. By directly comparing the result under Forced Update Rate of 5 to the result under Forced Update Rate of 132 above, it is found that the improvement is consistent. We see the improvements of the same range or slightly lower improvement for the WCDMA error, and for the DECT error patterns, the improvement is actually much larger under the Forced Update Rate of 5 condition.

| Error Pattern | WCDMA | | | | | | DECT | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 10 dB | | 20 dB | |
| | | | | | | | 14 km/h | 1.4 km/h | 14 km/h | 1.4 km/h |
| Improvement of luminance PSNR (dB) | +3.94 | +2.73 | +3.28 | +0.79 | +0.76 | +1.17 | +5.49 | +3.26 | +4.28 | +1.71 |

Table 2. Improvement of luminance PSNR (Annex V vs. Baseline) under Forced Update Rate of 5 (averaged over all sequences).

By simulations under the common conditions recommended by the ITU documentation, it is shown that Annex V, the syntax of Data Partitioning structure with RVLCs, brings a significant and consistent improvement of PSNR over the Anchor mode for H.263+ bitstreams sent through simulated error-prone channels. And the improvement holds also at different Forced Update Rate [4-8].

# IV. Conclusions

It has been well established that data partitioning will enhance error resilience performance. This is evidenced in the inclusion of data partitioning in various codecs, including MPEG-4, GA HDTV systems, as well as shown here in Annex V of H.263++. In H.263, the challenge was to propose a framework for data partitioning, to find a way to utilize robust entropy codes within this framework, and ensure that this approach led to consistent qualitative and quantitative improvements in the image quality. The work described in this paper accomplished these objectives, and has been adopted within the ITU standards.

# V. References

**[1]** ITU-T Recommendation H.263 (1995): “Video CODEC for narrow telecommunications channels at < 64 kbit/s”.

**[2]** J. Wen, J. Villasenor, and H.D. Shin. “Proposal for Single-Thread Motion Vector Prediction in H.263L Error Resilient Mode”, ITU-T Documentation Q15-C-35 (Eibsee, Germany 1997).

**[3]** J. Wen, J. Villasenor, and H.D. Shin. “Proposal for Error Resilience in H.263L”, ITU-T Documentation Q15-C-36 (Eibsee, Germany 1997).

**[4]** A. Li, J. Wen, J. Villasenor, and J.H. Park. “Model Implementation of Error Resilience Mode in H.263+ codec”, ITU-T Documantition Q15-E-19 (Whistler, Canada 1998)

**[5]** A. Li, M. Fong, J. Wen, J. Villasenor, and J.H. Park. “Test Results of Error Resilience with Modified Error Resilience Syntax with Data Partitioning and RVLC”, ITU-T Documentation Q15-C-20 (Whistler, Canada 1998)

**[6]** M. Luttrell, A. Li, M. Fong, J. Villasenor, and J.H. Park. “Simulation Results for Modified Error Resilience Syntax with Data Partitioning and RVLC”, ITU-T Documentation Q15-F-29 (Seoul, Korea 1998)

**[7]** M. Luttrell, J. Villasenor, J.H. Park, and D.S. Park, “Additional Simulation Results for Data-Partitioning-Based Error-Resilient Video Coding”, ITU-T Documentation Q15-G-12 (Monterey, USA 1999)

**[8]** J. Kim, K. Wuttipong, and C.J. Kuo, “Cross-validation of proposed data partitioning annex for enhanced error resilience”, ITU-T Documentation Q15-G-23 (Monterey, USA 1999)

**[9]** S. Kittitornkun, Y.H. Hu, and A. Li, “Results of cross-validation experiments for data partitioning”, ITU-T Documentation Q15-G-24 (Monterey, USA 1999)

**[10]** G. Sullivan, “Meeting Report of the Seventh Meeting (Meeting G) of the ITU-T 15/16 Advanced Video Coding Experts”, ITU-T Document Q15-G-48 (Monterey, USA 1999)

**[11]** J. Villasenor and D.S. Park, “Proposed Draft Text for the Annex V in H.263 for Determination at the SG meeting”, ITU-T Document Q15-I-14 (Red Bank, USA 1999)

**[12]** J. Wen, and J. Villasenor, “Reversible Variable Length Codes for Efficient and Robust Image and Video Coding”, Proceeding of 1998 IEEE Data Compression Conference, p.471 (Snow Bird, UT, USA 1998).

**[13]** S. Wenger. “Common Conditions for Video Performance Evluation in H.324/M Error-Prone Systems”, ITU-T Documentation Q15-E-50 (Whistler, Canada 1998)

**[14]** S. Wenger. “Proposed Changes for Mobile Test Model”, ITU-T Documentation Q15-F-38 (Seoul, Korea 1998)