The H.264/AVC Video Coding Standard for the Next Generation Multimedia Communication

M. Mahdi Ghandi, *Member, IEEE*, and Mohammad Ghanbari, *Fellow, IEEE*
(Invited paper)

***Abstract*—The capacity of a communication channel is determined by its bandwidth and signal-to-noise ratio. For a digital user, these parameters determine the bit rate and the probability of error and so affect the achievable quality of service. In the recent multimedia communication systems, bandwidth is still a limiting factor. Hence, effective video compression techniques are essential to reduce the amount of video data. The new H.264/AVC video coding standard jointly presented by the ITU-T and ISO/IEC experts groups has achieved a significant improvement in compression performance compared to the previous standards. It promises to deliver good quality of video at low bit rates. However if the probability of error is more than normal values, receiving the correct data is not guaranteed. This implies a need for useful operation of the video compression algorithms in error prone environments. This article presents an overview of the H.264/AVC features in terms of video compression techniques and error resilient coding. Some simulation results are provided to demonstrate the efficiency of the codec.**

***Index Terms*—Multimedia communication, video compression standard, error resilient coding.**

I. Introduction

DIGITAL video compression techniques have played a key role in recent multimedia communications. The limitation of bandwidth in communication channels and storage media demands more efficient video coding methods. On the other hand, introducing new applications and advances in multimedia technology demands video coding methods to include more complex and advanced features. Therefore, standardization of the video compression techniques is essential [1].

The H.261 recommendation [2] was the first standard for video conferencing applications that was developed by the Video Coding Experts Group (VCEG) of ITU-T in late 1980s. The Motion Picture Experts Group (MPEG) of ISO/IEC then introduced MPEG-1 standard [3] in early 1990s for storage of video on CD-ROMs. The MPEG-2 (or H.262) [4] which is now the most widely used video coding standard was a joint project of ISO/IEC (MPEG) and ITU-T (VCEG). Later on the two groups separately published two new advanced standards namely H.263 [5] and MPEG-4 [6] for coding video at low bit rates. The progress of video coding standards is summarized in Fig. 1.

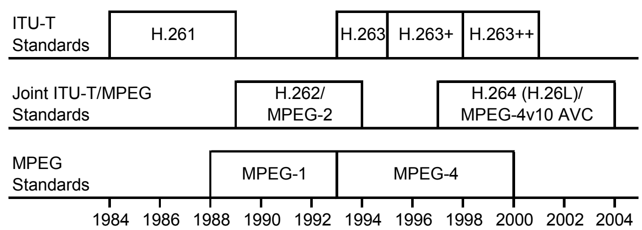

Fig. 1. Evolution of video coding standards ITU-T and MPEG

In early 1998 VCEG started a project named H.26L, with L standing for the long term objective to double the coding efficiency relative to the best of existing video coding standards. The H.26L project targeted at a wide variety of applications from low to high bit rates (e.g. multimedia over mobile networks and storage on optical devices respectively). The fruitful outcome of the MPEG-2/H.262 video codec product under the joint effort of the MPEG and VCEG encouraged the two experts groups in further collaboration. Therefore, in 2001 VCEG and MPEG formed a Joint Video Team (JVT) to finalize the project called by ITU-T as H.264 and by ISO/IEC as MPEG-4 part 10, Advanced Video Coding (AVC) [7], [8]. However for consistency, throughout this paper we call it just H.264.

Simulation results show that H.264 has achieved substantial superiority of video quality over that achieved by H.263++ and MPEG-4. The JVT test model software (a laboratory test model of H.264) has achieved up to 50 per cent in bit rate saving compared with the existing most optimized H.263 or MPEG-4 codec [9]-[11]. This means that H.264 offers significantly higher quality levels with the same bit rates.

As well as superior Rate-Distortion (RD) efficiency, H.264 can operate in low delay mode to adapt telecommunication applications, while allowing higher processing delay in applications with no delay constraints [12]. In the H.264 design there were efforts to keep the codec in a certain level of complexity [13]. The decoder in particular, has a reasonably low complexity that makes the advanced decoder appropriate for applications with limited processing power [14], [15].

The H.264 codec has a feature that conceptually separates

M. Ghanbari is a professor of video networking in the University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK. (Telephone: +44-1206-872434; Fax: +44-1206-972900; e-mail: ghan@ Essex.ac.uk).

M. M. Ghandi is a senior research officer in the video networking group of the University of Essex. (email: mahdi@essex.ac.uk)

the Video Coding Layer (VCL) from the Network Abstraction Layer (NAL). VCL provides the core compressed video contents, while NAL supports delivery over various types of network. This network friendliness feature of the standard facilitates easier packetization and better information priority control. In addition, to adapt H.264 to applications involving bit errors and packet losses, a number of error resilience techniques are provided in the standard.

This paper presents an overview on the H.264/AVC video coding method which is given in Section II. The new features that make the standard more efficient than the previous ones are particularly emphasized. Error resilient features of the codec are discussed in Section III as they are the most demanding features of the codec for error prone channels. Section IV provides a description on the non-normative R-D optimization method utilized in the JVT test model software which is a key feature of the success of the method. Selected simulation results verifying the R-D efficiency of the algorithm and error resilient techniques are demonstrated in Section V.

II. An Overview of the H.264/AVC Video Coding Method

Standard video coding methods including H.264 are based on three fundamental redundancy reduction principles, namely spatial and temporal redundancy reduction and entropy coding [16]. To carry out spatial redundancy reduction in *intraframes* (Fig. 2), transform coding is applied using Discrete Cosine Transform (DCT) on rectangular blocks followed by quantization and entropy coding. Temporal redundancy reduction is carried out through modeling the movements of objects in *interframes* (Fig. 6) by Motion Vectors (MVs) generated in the Motion Estimation (ME) process. In this section, a description of the H.264 features for intra-coding is first presented followed by a description of new inter-coding techniques.

A. Intraframe Coding

Although efficient video encoders mainly use interframe prediction, nevertheless use of intraframe coding for parts of the picture is necessary to prevent error propagation. However, intraframe coding generates a large bit rate, and hence in order for H.264 to be efficient, special attention is paid on intra-frame coding.

H.264 takes advantage of correlations between neighboring blocks to achieve better compression in intra coding. In this coder every intra 16x16 pixel Macroblock (MB) in a picture is first predicted in an appropriate mode from the already coded and reconstructed (R) samples of the same picture. As Fig. 2 shows, the difference between the predicted block (P) and the original one (prediction residual) is calculated, DCT transformed, quantized and entropy coded. The reconstructed samples are then filtered to generate the decoded frame and stored in a buffer to be used as a reference for coding of the future pictures.

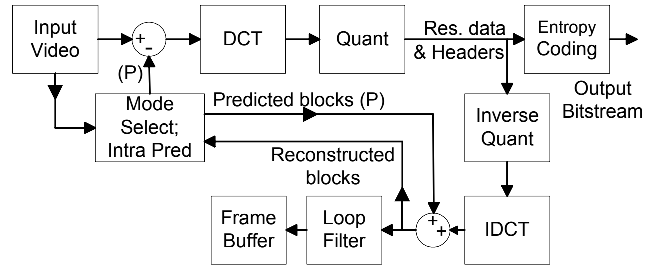

Fig. 2. A simplified block diagram of intraframe coding in the H.264 coder

1) Intra Prediction Modes: There are nine advanced prediction modes for luminance (luma) samples when the MB is partitioned into 4x4 blocks (i.e. intra-4x4 modes). Additionally, four other modes are used for predicting the whole 16x16 intra MB (intra-16x16). The chrominance (chroma) components can also be predicted in 4 different modes (i.e. intra-chroma modes). Various intra-4x4 modes are demonstrated in Fig. 3, where the prediction values for pixels are calculated from the neighboring boundary pixel values. Each mode is suitable to predict directional structures in the picture at different angles (e.g. horizontal, vertical, diagonal, etc.). Intra-16x16 and intra-chroma modes consist of horizontal, vertical, DC and plane modes. In these modes similar to the intra-4x4 modes, an arrangement of neighboring boundary pixels produce prediction pixels. For more details of each intra prediction mode, the H.264 recommendation should be consulted [7].

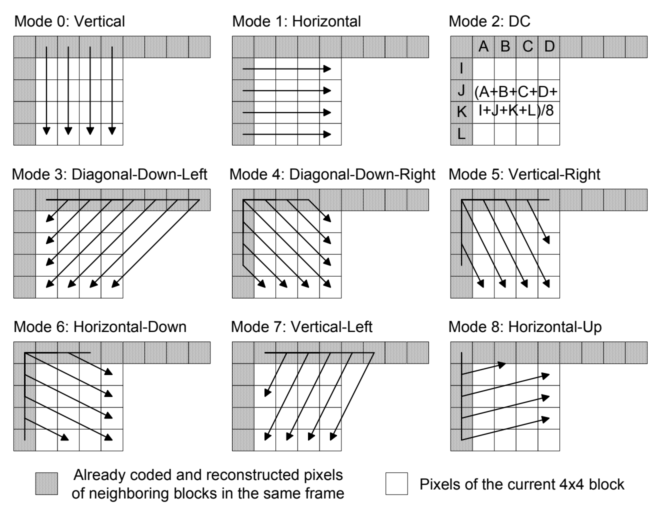

Fig. 3. Intra 4x4 prediction modes

2) Transform and Quantization: In H.264 similar to the other standards, a transformation and quantization is applied on the prediction residuals. However, H.264 employs a 4x4 integer transform as opposed to the 8x8 floating point DCT transform used in the other standard codecs. The transform is an approximation of the 4x4 DCT and hence it has similar coding gain to the DCT transform. Since the integer transform has an exact inverse operation, there is no mismatch between the encoder and the decoder which is a problem in all DCT based codecs.

$$T_{\text{int}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Fig. 4. The integer 4x4 forward transformation matrix used in the H.264 codec

After the transformation using the matrix of Fig. 4, each coefficient is scaled with a specified factor to make the final transform coefficient. The scaled coefficients are then quantized with a quantization step size determined by a given Quantization Parameter (QP). In H.264 the values of the quantizer step sizes have been defined such that the scaling and quantizing stages are mixed to be performed by simple integer operations in both encoder and decoder [17]. In particular, the inverse operations of scaling, quantizing and transforming are directly described in the standard using pure integer operations [7]. This can significantly reduce the processing power which is useful in power constrained processing applications, such as video over mobile networks and allow increased parallelism.

It should be mentioned that during the preparation of the standard, a variable block size transform method has been envisaged [18]. It includes 4x4, 4x8, 8x4 and 8x8 transforms and improves the efficiency of the codec. However, since it is not mature enough and adds more complexity to the codec, it has not been yet included in the first generation of the standard.

3) Entropy Coding: Before transmission, generated data of all types are entropy coded. H.264 supports two different methods of entropy coding namely Context Adaptive Variable Length Coding (CAVLC) and Context Adaptive Binary Arithmetic Coding (CABAC). As well as a conceptual difference between the two methods, CABAC is more efficient than CAVLC which itself is superior to the conventional VLC (Huffman) used in the other standard codecs.

a) CAVLC: When the codec is in this mode, the residual data is coded using CAVLC but other data are coded using simple Exp-Golomb codes which are listed in table I. These data are first appropriately mapped to the Exp-Golomb codes depending on the data type (e.g. MB headers, MVs, etc.), and then the corresponding code words are transmitted.

TABLE I
EXP-GOLOMB CODE WORDS

Code	Code Word
0	**1**
1,2	**01**0, **01**1
3,4,5,6	**001**00, **001**01, **001**10, **001**11
7,8,9,10,11,…	**0001**000, **0001**001, **0001**010, **0001**011, **0001**100, …
…	…

The zigzag scanned quantized coefficients of a residual block are coded using Context Adapting VLC tables. The already coded information of the neighboring blocks (i.e. upper and left blocks) and the coding status of the current block determine the context. Optimized VLC tables are specifically provided for each context to efficiently code the coefficients in different statistical conditions.

b) CABAC: In this mode, the generated data including headers and residual data are coded using a binary arithmetic coding engine. The compression improvement of CABAC is the consequence of non-integer length symbol assignment, adaptive probability estimation and improved context modeling scheme. In H.264, the CABAC is designed in a way that can be implemented using simple integer operations and table look-ups adapting it to low complexity applications [19].

A block diagram of CABAC coding process is depicted in Fig. 5. In order to code a syntax element, it is first mapped to a binary sequence called bin string. In the standard, proper binarization mapping schemes are provided for different types of data. For each element of the bin string (i.e. bin) a context index is defined based on the neighboring information and the coder status. There are 399 different contexts in the standard for various types of data, and the context modeling scheme (i.e. derivation of the context index) for each data type is clearly specified. The binary arithmetic coder engine then codes the bins using associated probability estimation tables addressed by the context index and generates the output stream. Subsequently, the probability tables are updated based on the coded bins for the future use.

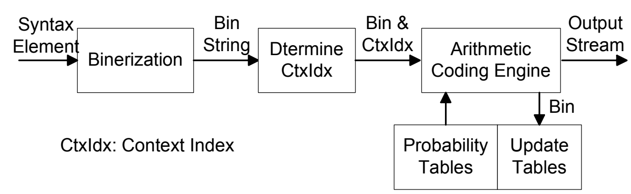

Fig. 5. A simplified block diagram of CABAC coder

4) In-Loop Deblocking Filter: To reconstruct a coded picture, it is filtered using an adaptive deblocking filter. The filter removes visible block structures on the edges of the 4x4 blocks caused by block-based transform coding and motion estimation [20]. The strength of filtering is adaptively controlled by the coded information such as QP and the block texture.

As well as subjective quality improvement, simulation results show that applying the filter in the encoder loop improves the compression efficiency of the codec [20]. Furthermore, if the filter is only applied in the decoder, an extra buffer in the decoder is needed to store the non-filtered frames to maintain synchronization with the encoder. These advantages of the in-loop filter were the reasons that it was accepted in the standard, despite an increase in the codec complexity.

B. *Interframe Coding*

In H.264, similar to its predecessor standards every MB of an interframe could be coded in intra or inter mode. If the intra mode is selected, the MB is coded as explained in Part A. Otherwise, the interframe prediction including ME, mode selection and Motion Compensation (MC) is performed to produce the predicted blocks. A block diagram of interframe coding in the H.264 coder is depicted in Fig. 6. Similar to intraframe coding, the residual data of original and predicted blocks, are transform coded. The inter prediction method in H.264 has some interesting features which are explained in the following parts.

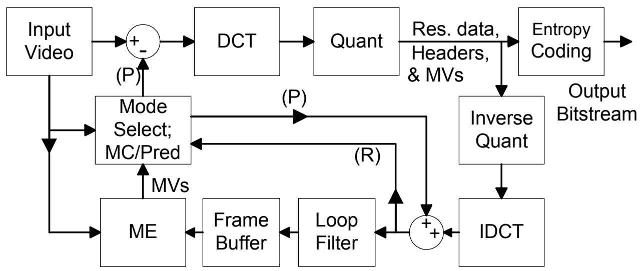

Fig. 6. A simplified block diagram of interframe coding in the H.264 encoder

1) Inter Prediction Modes: Interframe predictive coding is where H.264 makes most of its gain in compression efficiency. Motion compensation on each 16x16 MB can be performed with various block sizes and shapes illustrated in Fig. 7. The partitioning choice of a MB into 16x16, 8x16, 16x8 or 8x8 blocks is determined by *mb-type*. In 8x8 mode (i.e. *mb-type* 3) each of the blocks can be further divided independently into 8x8, 8x4, 4x8 or 4x4 sub-partitions determined by *sub-mb-type*. Note that each of these blocks contains its own MV and hence more precise motion compensation can be performed when the MB is divided into smaller blocks.

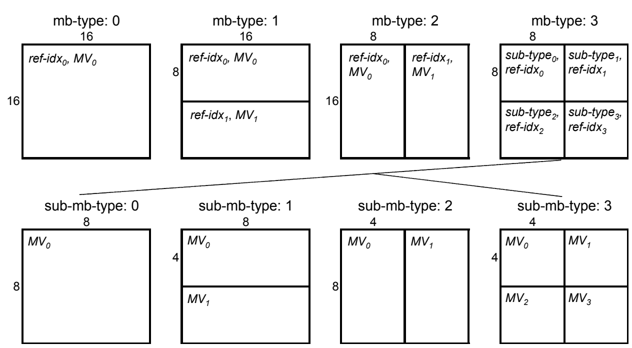

Fig. 7. Top: various 16x16 MB partitioning modes for MC. Bottom: sub partitioning modes of 8x8 blocks when mb-type 3.

2) Multiple Reference Prediction: The H.264 standard offers the option of using many previous pictures for prediction. Every MB partition (but not sub-partition) shown on the top part of Fig. 7, could have a different reference picture that is more appropriate for that particular block. This will increase the coding efficiency and produce better subjective quality. Moreover, using this mode improves the robustness of the bitstream to channel errors [12].

3) Quarter Sample MVs: The objects movements in the consecutive pictures of a sequence particularly for reduced size pictures such as QCIF, are not necessarily in integer pixel units. Therefore, to improve the motion modeling scheme of inter pictures, the MVs are in quarter sample precision. To generate the values of half-pixel positions a 6-tap Finite Impulse Response (FIR) filter is applied to integer position samples. The quarter-pixel samples are then generated using simple interpolation between neighboring (integer or half-pixel position) samples [21].

4) B-pictures consideration: B-pictures are encoded using both past and future pictures as references in contrast to P-pictures which are only predicted form the past references. In other words, each block of a B-picture can be predicted from either of two reference blocks or a linear combination of them. In H.264 these references could be both in the past or one in the past and one in the future. New concept of *direct* prediction mode is specified in H.264, where no data (such as MVs and reference indices) is present in the bitstream for the prediction process and they are derived from the available data of co-located MBs of the subsequent pictures. To support B-picture coding as well as P-picture, proper mode referencing tables and entropy coding methods are specified [7][22].

III. H.264 NAL Layer and Error Resilience Features

In a communication channel the quality of service is affected by the two parameters of bandwidth and the probability of error. Therefore, as well as video compression efficiency which is provided through the VCL layer, the adaptation to communication channels should be carefully considered. The concept of NAL layer and the error resilience features in H.264 are to provide an appropriate VCL representation for conveyance on a variety of channels to cope with their erroneous situations.

A. *NAL*

The Network Abstraction Layer facilitates the delivery of the H.264 VCL data to the underlying transport layers such as RTP/IP, H.32X and MPEG-2 systems [23]. Each NAL unit could be considered as a packet that contains an integer number of bytes including a header and a payload (see Fig. 8). The header specifies the NAL unit type and the payload contains the related data.

TABLE II
NAL unit types

NAL unit type	Class	Content of NAL unit
0	-	Unspecified
1	VCL	Coded slice
2	VCL	Coded slice data partition A
3	VCL	Coded slice data partition B
4	VCL	Coded slice data partition C
5	VCL	Coded slice of an IDR picture
6-12	Non-VCL	Supplemental Information, Parameter Sets, etc.
12-23	-	Reserved
24-31	-	Unspecified

Table II gives a summarized list of different NAL unit types. NAL units 1 to 5 contain different VCL data that will be described later. NAL units 6 to 12 are non-VCL units containing additional information such as parameter sets and supplemental information. Parameter sets are header data that are unchanged in a number of NAL units, and then are sent to prevent repeating them. Supplemental information is timing or other addressing data that enhances the decoder usability but are not essential in decoding the pictures. NAL units 12 to 23 are reserved for future use of H.264 extensions and the types 24 to 31 are available for use by different applications.

B. Error Resilience Features

H.264 provides features that make the generated bitstream robust to the bit errors and packet losses. Following is a brief description of each of these features:

1) Resynchronization (Slice) Headers: Each frame can be divided into several slices; each contains a flexible number of MBs. In each slice the arithmetic coder is aligned and the predictions are reset. Hence, every slice in the frame is independently decodable. Therefore, they can be considered as resynchronization points that prevent propagation of a probable error to entire frame. If the resynchronization markers used too often, they limit the damaged area more tightly, so improving the error resilience of the codec. It is obvious that the slicing introduces some overhead and reduces the compression efficiency, but in the erroneous situations, due to the better error resilience, the received video quality could be much better.

In H.264, each slice is placed in a separate NAL unit (see table II). The slices of an IDR picture (i.e. a picture with all intra slices) are located in type 5 NAL units, while those belonged to a non-IDR picture are placed in NAL units of type 1 to 4 depending on the Data Partitioning (DP) mode which is explained in the following part.

2) Data Partitioning: DP is another important and efficient way to make a bitstream more robust. It is proposed due to the fact that the symbols appeared earlier in the bitstream suffer less from the errors than those which come later. Therefore, by bringing the more important parts of the video data (such as headers and MVs) ahead of the non-important data, the channel error side effect can be significantly reduced.

In H.264, when DP is enabled, every slice is divided into three separate partitions (Fig. 8) and every partition is located in a particular NAL unit (see table II as well). Therefore, data partitioning can be used as an efficient layering method that separates the data with different importance.

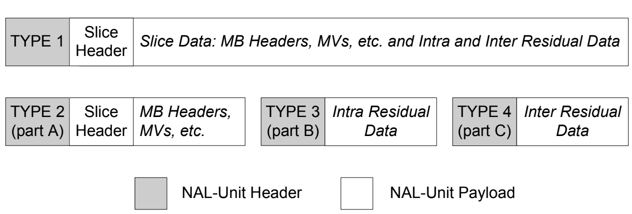

Fig. 8. A non-IDR slice placed in NAL units. Top: data partitioning is disabled. Bottom: data partitioning is enabled.

By partitioning the data into different NAL units, applying unequal error protection to various important parts of video data becomes straightforward. For example a high error protection can be applied to NAL units of type 2, 3 and 5 (headers, intra residuals and IDR pictures respectively) and a lower protection for type 4 units (inter residual data), since they are less important. It should be noted that there is no DP for IDR pictures. However, since they have a different NAL unit type (type 5), special error protection could be applied to them.

3) Flexible MB Ordering (FMO): In FMO mode, MBs are allowed to be assigned to any slice in a frame, so they are flexible to be transmitted in a non-scanning order. Since each slice is independently decodable, one can mange to spatially interleave MBs into different slices. Therefore, if one of the slices is missed, its missing MBs are surrounded by correctly received MBs. Hence, by applying an appropriate error concealment method, the visual artifact of the losses could significantly be reduced.

4) Redundant Slices: H.264 has a new feature that allows the encoder to send Redundant Slices (RSs) containing information of the same primarily transmitted MBs. If the original slices are lost, the RSs represent an alternative data (but somewhat in lower quality) that can be used to recover the corrupted MBs.

5) Intra Refresh and Multiple References: Features such as Intra modes and Multiple Reference Selection (MRS), improve the error robustness of a bitstream as well as the encoding efficiency. The intra MBs do not use temporal prediction, so they prevent error propagation through the bitstream. Similarly, using multiple references (especially in the presence of a back-channel) can prevent corrupted pictures to be used as references and hence stop the error propagation [24],[25].

Intra MBs can be inserted very often in the bitstream and in this case the coding efficiency is reduced, but the error robustness is improved. This is the case for MRS mode which the encoder can be purely optimized for coding efficiency or to generate more robust bitstreams.

IV. Rate-Distortion Optimization

In H.264 similar to the other standards, only the bitstream syntax and the decoding procedure are specified, and the encoder is flexible to have different implementations. It should select the parameters such as MB modes, MVs, QP and the residual data for each MB among several choices. This huge amount of choices makes optimization of the encoder very crucial to achieve a good R-D efficiency. In other words, a poor design of encoder can generate even lower quality bitstreams than a simple traditional video coder. In the JVT reference software a Lagrangian optimization technique is applied to achieve R-D efficiency [9].

A. Lagrangian Optimization Technique

Lagrangian techniques are based on converting a constraint optimization problem to an unconstrained one [26]. In (1) the problem is to minimize the distortion (D) with a constraint that the rate (R) should be less than R_c:

$$\min D \big| R < R_c \tag{1}$$

Equation (1) is converted to (2) where the problem is to minimize the Lagrangian cost J, with λ being the Lagrangian parameter:

$$\min J \big| J = D + \lambda \times R \tag{2}$$

The selection of encoding parameters for every MB in a picture determines its rate (r) and distortion (d) and the sum of these values generates R and D. Assuming that r and d of a particular MB are only dependant on its coding parameters (and not the others,) the optimization of (2) is simplified to minimizing the cost of each MB separately:

$$\min j \big| j = d + \lambda \times r \tag{3}$$

B. Optimization Process

In the JVT software, motion estimation and mode selection have been optimized using Lagrangian method. The ME process finds MVs that minimize j_{mv}, calculated using the bits needed to send each MV (r_{mv}) and the relating Sum of Absolute Distortion (SAD_{mv}):

$$j_{mv} = SAD_{mv} + \lambda_{Motion} \times r_{mv} \tag{4}$$

$$j_m = SSD_m + \lambda_{Mode} \times r_m \tag{5}$$

After determining the optimum MVs for inter modes, the Lagrangian cost of each (intra and inter) mode (j_m) is calculated using (5), where the required number of bits to send all MB information is $r_{m,}$ and SSD_m is the Sum of Squared Differences (SSD) between the original and the reconstructed pixels. The mode that has the lowest j_m among all others is selected as the optimum one. Note that in intraframes only the intra modes are allowed and searched while in interframes, both types are examined.

C. Selection of λ

In the above optimization process, the value of λ is calculated using an empirical formula of (6):

$$\begin{aligned} \lambda_{Mode} &= 0.85 \times 2^{(QP-12)/3} \\ \lambda_{Motion} &= \sqrt{\lambda_{Mode}} \end{aligned} \tag{6}$$

This relationship between QP and λ is extracted through experiments similar to what is described in [27] for H.263. In these experiments the average selected values of QP by optimization process with various given λ values are calculated and equation (6) has been established. Fig. 9 gives the results of the experiment for the Foreman test sequence. The encoder examines different values of QP as well as MB modes and selects the QP that has the minimum Lagrangian cost. From the figure it is clear that when λ is calculated using (6) by an specific QP, the average number of selected QP is equal to that particular QP, and this verifies the accuracy of relationship of (6).

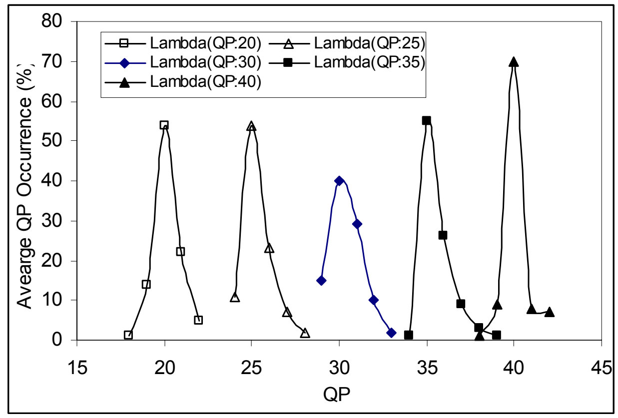

Fig. 9. Average QP occurrence for different λ values calculated form (6), Foreman QCIF@10Hz

Therefore, for coding pictures a QP is first selected and a λ value is then calculated to precede the optimization process. It can be seen that in this method the bit rate is not directly controlled and the resulting bitstream is variable bit rate. To add the rate control feature to the software there are some proposals in the literature [28]-[31]. In addition, since all possible modes of every MB are examined, this method is a time consuming process. Some proposals have simplified the process with an acceptable delay [32]-[35].

V. Selected Simulation Results

In this section, through various simulations we intend to demonstrate the two fundamental properties of an H.264 codec namely its compression efficiency and its error resilience property.

A. R-D Efficiency

Four different video test sequences have been selected in our simulations namely: Foreman, News, Mobile and Akio, each have a different texture and movement property. The sequences were coded by JM7.6 JVT reference software [36]. In the default settings for each test, the picture resolution is QCIF, the frame rate is 10Hz, number of reference frames is 1, ME search window is 16, the entropy coding method is CABAC, Lagrangian R-D optimization is enabled, and bitstreams contain an Intra picture followed by 99 P-pictures.

Fig. 10 demonstrates the results of these four tests in the described default modes. The results clearly show that the R-D efficiency (or quality: PSNR) of the codec is strongly dependant on the contents of the pictures. To evaluate the coding features, we have performed a series of tests. In each test, settings that are unique for that test and are different from the defaults will be clearly specified. For briefness, we will just demonstrate two extreme cases (i.e. the tests that have the best and the worst results) for each series.

1) Intra Prediction Modes: To evaluate the efficiency of intra prediction modes adapted in H.264, we have selected 10 frames of each sequence, with 10 frames distance between each, and coded them in Intra mode and their averages are calculated. Each test was done in three scenarios; firstly all intra prediction modes were allowed, secondly they were limited to 16x16 intra modes only, and finally the encoder was forced to select 4x4 DC mode only. In the latter scenario, we have amended the bitstream structure in a way that no header is sent for intra prediction mode addressing, since there is only one mode.

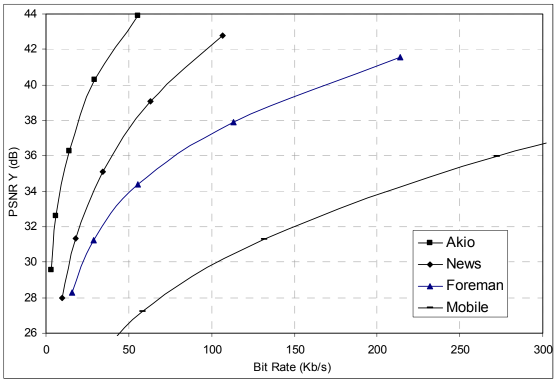

Fig. 10. R-D curves of the selected tests in default modes.

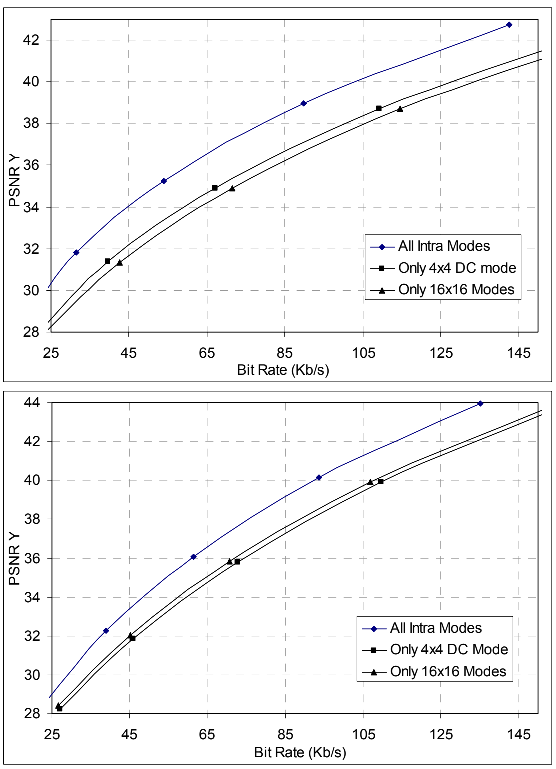

Fig. 11. All Intra pictures, 3 Hz, Top: Foreman, Bottom: News

Fig. 11 shows the results for the Foreman and News sequences where it is clear that the advanced intra prediction modes of H.264 have a significant improvement in intra coding.

2) CABAC: The next test is to evaluate the improvement caused by the used entropy coding algorithm. In Fig. 12 the R-D (quality) curves of the Foreman and Akio tests are compared in CABAC and CAVLC modes. In these tests, the CABAC entropy coding mode has outperformed the CAVLC method by up to 8%. Obviously the achieved saving is dependant on the test sequence and the bit rate as well.

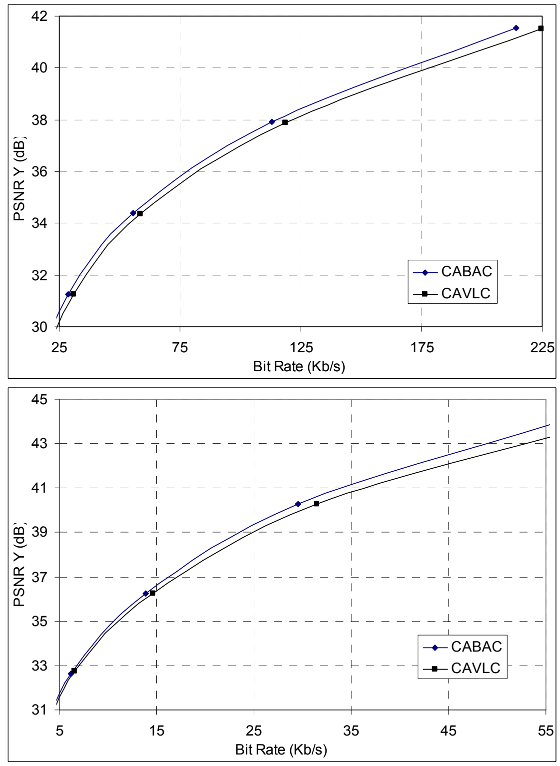

Fig. 12. CABAC and CAVLC modes, Top: Foreman, Bottom: Akio

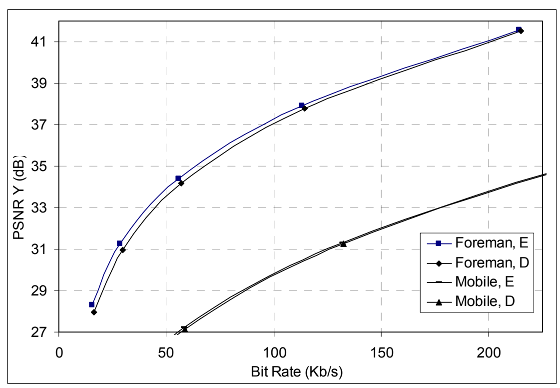

Fig. 13. Foreman and Mobile tests, deblocking filter enabled and disabled

Fig. 14. Foreman, QP: 40, Frame 1, Filter disabled (left) and enabled (right)

3) Deblocking Filter: As mentioned earlier, the use of deblocking filter in the encoder loop will improve the R-D efficiency as well as the subjective quality. Fig. 13 shows that

how applying the deblocking filter has improved the R-D efficiency for the Foreman test, especially in lower bit rates. On Mobile test due to the complex texture and high bit rate, in-loop filter has not improved the efficiency. Fig. 14 demonstrates the subjective quality of the Foreman picture when the filter is enabled and disabled. It is clear that the deblocking filter has subjectively improved the picture quality by smoothing the blocking artifacts.

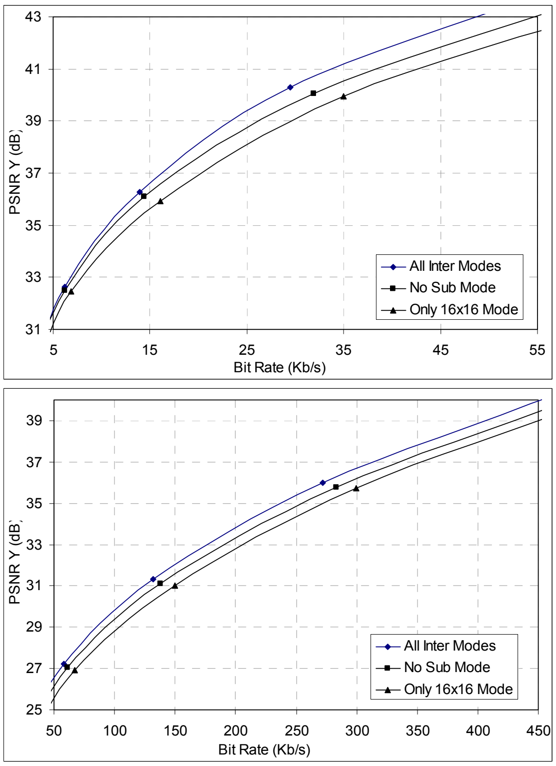

Fig. 15. Top: Akio test, Bottom: Mobile test. (i) all inter prediction modes are allowed, (ii) only 16x16 to 8x8 modes are allowed and (iii) only 16x16 mode is allowed.

Fig. 16. MB inter partitioning modes, Foreman frame 2, high bit rate QP: 20 (left) and low bit rate QP: 40 (right)

4) Inter Prediction Modes: We have compared the performance of prediction modes under various scenarios of: (i) all inter modes are allowed (normal H.264 case), (ii) only 16x16 to 8x8 modes are allowed and (iii) only 16x16 inter mode is allowed. For the second and the third scenarios, we have amended the bitstream semantics in a way that no *sub-mb-mode* is sent and for the 16x16 scenario, there is no *mb-mode* either (since there is only one mode). Fig. 15 shows the results for Akio and Mobile. It can be seen that the use of all H.264 inter prediction modes significantly improves the coding efficiency. In fact this feature of H.264 is one of the key reasons of it success. Note that in lower bit rates the improvement caused by the sub-MB-modes is less than the higher rates. The reason is that when the bit budget is very limited, the R-D optimization process selects 16x16 modes mostly rather than smaller division modes. Fig. 16 shows the overlay of MB sizes for the high and low bit rate of the second frame of the Foreman test sequence.

5) Multiple Reference Prediction: Fig. 17 shows the R-D (quality) curves of the Foreman and News tests when the number of reference frames for inter prediction is varied from 1 to 9. We have removed some middle values to make the graphs more readable. From the figure is can be observed that having more references for prediction improves the coding efficiency. However, it can be seen that when the movement between frames is not significant (like the News test,) there is not much to gain from multiple reference, since there are little differences between the reference pictures. Furthermore, note that when the encoder searches for the best match in more references, the encoding delay will increase. For example in 9-frame reference scenario, the ME process is 9 times slower than 1-frame reference scenario, and hence the overall coding delay will significantly grow.

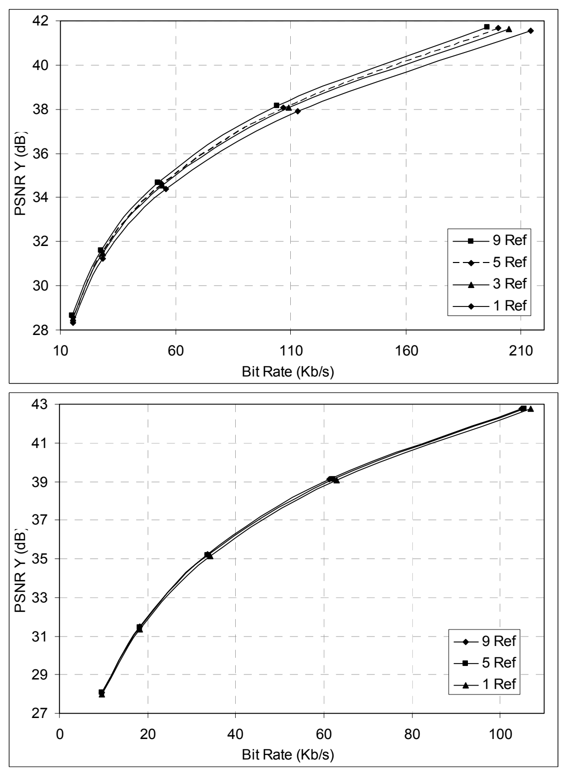

Fig. 17. Top: Foreman, Bottom: News. Different number of prediction reference frames.

B. Evaluation of Error Resilience Features

To be able to simulate the impact of channel errors, we have modified the JVT test model decoder software to cope with errors. To introduce the channel noise on a bitstream, a discrete two-stage Elliot-Gilbert model is used [37]. Based on the channel bit error rate and the mean burst length the model randomly alters the polarity of the bits (0 or 1). During the decoding procedure, when the first error in a slice is detected, the decoder skips the bits to the next slice and marks un-decoded MBs as corrupted.

When a picture is completely decoded, the corrupted MBs are concealed using their correctly received neighboring MBs. In the concealment method [38], the MVs are recovered using a boundary matching technique. Finally, the quality of the decoded and concealed pictures is assessed based on their PSNR. To make the statistics more reliable, every simulation is run 30 times and the resulting distortions are averaged.

1) Data Partitioning: In Fig. 18 the average Luma PSNR of the foreman coded bitstream with different bit error rates is illustrated. Bit rates of the bitstreams were set to 100 kb/s, using our Lagrangian rate controller [28]. From the figure it can be observed that enabling the DP which is simple with negligible overhead (and hence, small quality degradation) has significantly improved the resilience of the codec to channel errors.

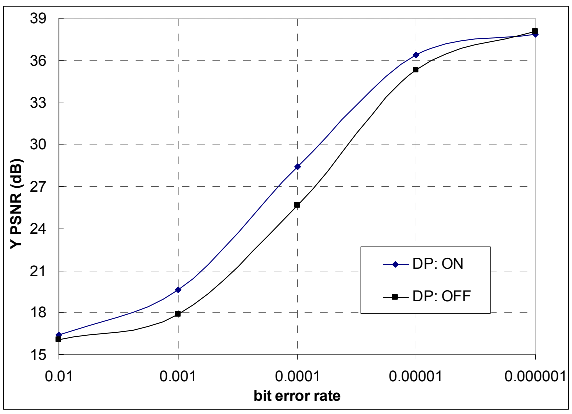

Fig. 18. PSNR vs. bit error rate when DP is enabled and disabled. Foreman, 100 KBits/Sec.

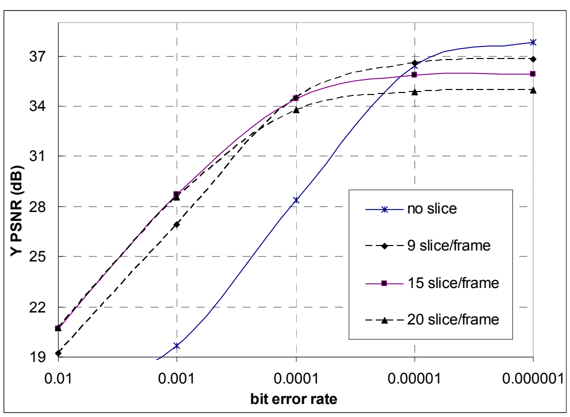

Fig. 19. PSNR vs. bit error rate, DP enabled, with and without slices. Foreman, 100 KBits/Sec.

2) Slice Structure: Fig. 19 shows the quality of the decoded erroneous video bitstream of the same test sequence when the slice structure is enabled. Due to the insertion of more resynchronization headers (i.e. more slices per frame) higher overhead is used and hence, the video quality in the error free situation is degraded. However, in higher bit error rates, the sliced bitstreams have significantly better qualities. It should be noted that when the number of slices is more than a specific amount (in this test 15), no further improvement is achieved.

3) Unequal Error Protection: Fig. 20 shows the quality of the received video when the DPA part of the data has error protection and supposed to be error free. This is possible by applying advanced channel coding techniques [39]. In this particular test, DPA is 40% of the bit rate. Additionally, the applied channel coding technique introduces a 25% overhead. Hence, to make the comparison fair, we have adjusted the non-protected bitstream bit rate by 10% more than the protected one. From the figure it is clear that by applying error protection to the DPA, the average output quality in high error rates has been dramatically improved.

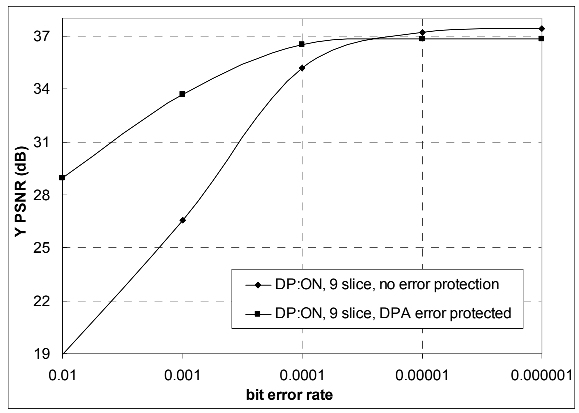

Fig. 20. PSNR vs. bit error rate, DP enabled, 9 slices per frame, with and without DPA error protection. Foreman 100 and 110 kb/s respectively.

VI. Conclusion

The H.264/AVC video coding standard has achieved a significant improvement compared to its predecessors. As well as new features that improves the compression efficiency such as advanced inter and intra prediction, H.264 supports a number of error resilience techniques that facilitate the codec to cope with different channel situations. These characteristics of AVC make it an ideal codec for applications with very limited channel capacity and extremely error prone channels such as mobile systems and video telephony. Due to its high compression efficiency, the codec can be used for coding of high quality video at lower rates. Therefore, this standard will be a serious contender for a variety of next generation multimedia applications. For instance, the DVD Forum Steering Committee has recently selected H.264 decoding as a mandatory capability for players of its upcoming new HD-DVD format. This is the first selection of the new standard by a major industry consortium for a consumer end-user product technology with clear potential for extremely widespread use.

Acknowledgment

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC) of the UK.

REFERENCES

[1] Mohammed Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding.* London, UK: IEE, 2003, ch. 1.

[2] "Video Codec for Audio Visual Services at px64 kbits/s," ITU-T Recommendation H.261, 1990

[3] "Coding of Moving Pictures and Associated Audio for Digital storage Media at up to About 1.5 Mbits/s," ISO/IEC 1117-2: Video (MPEG-1), November, 1991

[4] "Generic Coding of Moving Pictures and Associated Audio Information," ISO/IEC 13818-2: Video (MPEG-2), May 1996

[5] "Video Coding for Low Bit Rate Communication, Version 1," ITU-T Recommendation H.263, 1995

[6] "Coding of Audio-Visual Objects, Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 visual version 1), 1999

[7] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 144496-10 AVC)," Joint Video Team of ISO/IEC and ITU-T, March 2003

[8] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard and Ajay Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 560-576.

[9] Thomas Wiegand, Heiko Schwarz, Anthony Joch, Faouzi Kossentini and Gary J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 688-703.

[10] Daniel Alfonso, Daniele Bagni, Danilo Pau and Antonio Chimienti, "A Performance analysis of H.264 video coding standard," in *Proc. 23rd Picture Coding Symposium*, Saint-Malo, France, April 2003, pp. 23-28.

[11] Anthony Joch, Faouzi Kossentini and Panos Nasiopoulos, "A Performance analysis of the ITU-T draft H.26L video coding standard," Proc. 12th Int. Packet Video Workshop, Pittsburgh, PA, April 2002

[12] Mohammed Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding.* London, UK: IEE, 2003, ch. 9.

[13] Henrique S. Malvar, Antti Hallapuro, Marta Karczewicz and Louis Kerofsky, "Low-complexity transform and quantization in H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 598-603.

[14] Ville Lappalainen, Antti Hallapuro, and Timo D. Hamalainen, "Complexity of optimized H.26L video decoder implementation," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 717-725.

[15] Michael Horowitz, Anthony Joch, Faouzi Kossentini and Antti Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 704-716.

[16] Mohammed Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding.* London, UK: IEE, 2003, ch. 3.

[17] Iain E.G. Richardson, *H.264 and MPEG-4 Video Compression, Video Coding for Next Generation Multimedia.* Chichester, UK: John Wiley & Sons Ltd, 2003, ch. 6.

[18] Mathias Wien, "Variable block-size transforms for H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 604-613.

[19] Detlev Marpe, Heiko Schwarz and Thomas Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 620-636.

[20] Peter List, Anthony Joch, Jani Lainema, Gisle Bjontegaard and Marta Karczewicz, "Adaptive deblocking filter," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 614-619.

[21] Thomas Wedi and Georg Musmann, "Motion- and aliasing-compensated prediction for hybrid video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 577-586.

[22] Markus Flierl and Bernd Girod, "Generalized B pictures and the draft H.264/AVC video-compression standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 587-597.

[23] Stephan Wenger, "H.264/AVC over IP," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 645-656.

[24] Thomas Stockhammer, Miska M. Hannuksela and Thomas Wiegand, "H.264/AVC in wireless environment," IEEE Trans. Circuits Syst. Video Technol., vol. 13, July 2003, pp. 657-673.

[25] Bernd Girod and Niko Farber, "Feedback-based error control for mobile video transmission," Proceedings of the IEEE, vol. 87, No. 10, October 1999

[26] Antonio Ortega and Kannan Ramchandran, "Rate-distortion methods for image and video compression," IEEE signal processing magazine, vol. 15, Nov. 1998, pp. 23-50.

[27] Gary J. Sullivan and Thomas Wiegand, "Rate-distortion optimization for video compression," IEEE signal processing magazine, vol. 15, Nov. 1998, pp. 74-90.

[28] M. Mahdi Ghandi and Mohammed Ghanbari, "A Lagrangian optimized rate control algorithm for the H.264/AVC encoder," IEEE International Conference on Image Processing (ICIP), October 2004.

[29] Siwei Ma, Wen Gao, Feng Wu and Yan Lu, "Rate control for JVT video coding scheme with HRD consideration," in *Proc. Int. Conf. Image Processing,* Spain, Sep. 2003, pp. 793-796.

[30] Simone Milani, Luca Celetto and Gian Antonio Mian, "A rate control algorithm for the H.264 encoder," in *Proc. 6th baiona Workshop Signal Processing Comm.,* Spain, Sep. 2003.

[31] Zhengguo Li, Feng Pan and Keng Pang, "Adaptive Basic Unit Layer Rate Control for JVT," JVT of ISO/IEC MPEG & ITU-T VCEG, Pattaya, Thailand, 7-14 March, 2003

[32] Chuan-Yu Cho, Shiang-Yang Huang and Jia-Shung Wang, "An embedded merging scheme for H.264/AVC motion estimation," in *Proc. Int. Conf. Image Processing,* Spain, Sep. 2003, pp. 909-912.

[33] Peng Yin, Hye-Yeon Cheong Tourapis, Alexis Michael Tourapis and Jill Boyce, "Fast mode decision and motion estimation for JVT/H.264," in *Proc. Int. Conf. Image Processing,* Spain, Sep. 2003, pp. 853-856.

[34] Feng Pan, Xiao Lin and Rahardja, "Fast mode decision for intra prediction," JVT of ISO/IEC MPEG & ITU-T VCEG, Pattaya, Thailand, 7-14 March, 2003.

[35] K. P. Lim, S. Wu, D. J. Wu, S. Rahardja, X. Lin, F. Pan and Z. G. Li, "Fast inter mode selection," JVT of ISO/IEC MPEG & ITU-T VCEG, San Diego, USA, 2-5 Sep., 2003.

[36] JVT, "Test model software," http://bs.hhi.de/~suehring/tml/download

[37] Mohammed Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding.* London, UK: IEE, 2003, Appendix E.

[38] Ye-Kui Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabbouj, "The Error Concealment Feature in the H.26L Test Model," in *Proc. Int. Conf. Image Processing,* Sep. 2002, pp. 729-732.

[39] Peter Sweeney, *Error Control Coding*, Chichester, UK: John Wiley & Sons LTD, 2002