ERROR-RESILIENT TECHNIQUES FOR VIDEO TRANSMISSION OVER WIRELESS CHANNELS

S. Dogan, A. H. Sadka and A. M. Kondoz
Centre for Communication Systems Research (CCSR)
University of Surrey, Guildford GU2 7XH, Surrey, UK

e-mail: {S.Dogan, A.Sadka, A.Kondoz}@eim.surrey.ac.uk

Abstract—Video communications over wireless networks have always been a challenging issue. When a video stream is conveyed through a wireless communications channel to the receiver end, it is exposed to high levels of burst and random bit errors. These channel errors severely degrade the service quality of video communications, in particular for carrying out two-way communications, due to the propagation and accumulation of errors throughout the transmitted video stream. There are several procedures used to reduce these deteriorating effects, including the use of forward error correction techniques, feedback algorithms, error concealment methods and error resilience schemes. This paper presents these remedial solutions and describes popular error control and resilience algorithms, namely Turbo codes, error-resilient entropy codes and two-way decoding with reversible codewords. The improvement in perceptual quality offered by these schemes is demonstrated. This paper also shows that the combination of these three error control and resilience methods gives a better performance than conventional forward error correction techniques when compressed video is subjected to error-prone environments.

Index Terms—Video coding, wireless channel effects, error resilience.
I. INTRODUCTION

The rapid improvement and deployment of multimedia communications all over the world has been greatly expected over the last decade. As we are standing at the doorway of a new millennium, the researchers’ dreams of wide varieties of multimedia applications are about to come true. Multimedia communications comprise a broad range of applications; from speech, audio, video and image coding to text and graphics data communications. This paper focuses, particularly, on the video coding part of the multimedia applications, which has received considerable interest in recent years.

The integration of moving video as an integral part of multimedia environment is technologically one of the most demanding tasks. Unlike data signals, video information contains redundancies in both time and space. Hence, the digital representation of visual information in its natural form generates a huge amount of data. On the other hand, bandwidth is a major bottleneck for multimedia communications, particularly, for the video transmissions. For instance, raw video data is always hungry for bandwidth. For example, a 176×144 quarter common intermediate format (QCIF) 4:2:0 colour video sequence requires over 1 Mb/s [1]. Applications running on wireless networks in particular suffer more from the bandwidth requirements and the channel constraints. Therefore, digital video compression is one of the key issues in video communications, enabling efficient interchange and distribution of visual information. Video compression techniques are exploited to remove the redundancies from the video allowing reduced bit rates and feasible transmission solutions. Thus, video compression has to be applied before the video stream is conveyed through to the band-limited wireless channels.

Video over wireless networks is a natural evolution originated by the high demand for accessing the multimedia information in mobile and remote environments. However, wireless channels have the inherent impacts which affect the entire data transmission, such
as high levels of noise in terms of bit-error-rate (BER) and interference, varying delay causing jitter effect, Rayleigh fading and multipath effects and shadowing [2, 3]. When the effects of limited transmission power associated with the mobile terminals and the limited bandwidth constraints are also considered, wireless communications become wide open to the research and development activities.

The raw video data is compressed by means of video encoders at the source end of the entire transmission environment to make it fit into the bandwidth-limited wireless channels. However, this process has a significant effect of reducing the robustness of the data against the high levels of channel errors, due to the removal of the redundancy from the video data. Therefore, when the compressed video data is conveyed to the receiver end through a wireless environment, it is subjected to channel errors in the form of burst and random bit errors [1]. These errors have highly destructive effects on the quality of service (QoS) of video in the case of non-error resilient transmissions. This paper discusses the reliable transmission issues of video data over wireless communication environments.

The paper is organised as follows. Section II gives a brief introduction to video coding. Section III explains the effects of wireless channel errors on video streams whilst Section IV discusses the possible solutions for robust video transmissions over wireless networks. Section V describes the proposed solutions and their implementation on video coders. Section VI presents the obtained results from the experiments and their discussion. Finally, Section VII concludes the paper.

II. VIDEO CODING

A number of video coding and compression algorithms have been developed so far. Amongst all, the most successful and the most well-known ones are H.261, H.263 and its annex extensions, such as H.263+ and H.263++, of International Telecommunications
Union (ITU), and Motion Picture Experts Group-v.1 (MPEG-1), MPEG-2 and MPEG-4 of International Standards Organisation (ISO). Both H.261 [4] and H.263 [5] are video-conferencing oriented low and very low bit rate video coders, respectively. MPEG-1 [6] is the standard for coding of moving pictures and associated audio for digital storage media at up to 1.5 Mb/s whilst MPEG-2 [7] is the standard for generic coding of moving pictures and associated audio at higher bit rates. Finally, MPEG-4 [8] is the ISO’s recently released very low to high bit rate, object oriented video communications standard with superior functionalities compared with its predecessors: MPEG-1 and MPEG-2 [9]. These various video coding algorithms exploit several different video coding techniques, such as vector quantisation, wavelet-based, sub-band coding or DCT-based techniques. This section, however, limits the discussion of video coding methods to the DCT-based techniques.

![A block-based video encoder block diagram.](image)

Video coding is incorporated by video compression which removes the redundancy from the raw video data stream. Video compression is a process whereby a collection of algorithms and techniques replace the original pixel-related information with more compact descriptions. When compressed data is received over a communications link, it must be possible to expand the data back by decompression. Decompression is the reverse process
of encoding, so that it decodes the descriptions back to pixels for display. At its best, video compression is transparent to the end user. A block diagram of a typical video compression scheme is depicted in Fig. 1.

The encoder handles its raw video input on a frame by frame basis. Each individual frame is processed using particular encoding scheme that has been selected by the encoder [10]. If frame encoding is performed using only the data from a single frame, an intra frame is produced. These intra frames allow a video stream to be accessed at random points. Compression is accomplished through frequency transformation, which provides a moderate bit rate reduction. The video is usually transformed from the spatial domain to the frequency domain using a discrete cosine transform (DCT) on 8×8 blocks of pixels. After quantisation, a DCT block usually contains numerous zero magnitude coefficients, and a few low frequency values. The coefficients are read in a zigzag order from the top left corner, creating a stream of values representing increasing frequencies. Run-length encoding is then employed to take advantage of the long runs of zeros within the stream. Thus, the coefficients are encoded as a series of run-amplitude pairs, representing a run of zeros and a preceding non-zero coefficient. Variable length coding, such as Huffman algorithm [11], is used to minimise redundancy in the binary output. It uses shorter codes for commonly occurring pairs, and longer codes for less common ones. Intra frames contain no motion data as they are encoded with reference only to themselves.

The content of inter frames, sometimes called predictive frames, is dependent upon a prediction taken from a certain number of reference frames. Each frame is divided into square blocks of 16×16 pixels. These blocks are called macroblocks (MBs). Motion estimation is employed to reveal a motion vector corresponding to the relative position of a similar block in the reference frame. Estimation is performed within a search window around the MB as shown in Fig. 2. Motion compensation is then performed using the
computed motion vectors. A comparison is made between the motion compensated frame and the original image to find differences in intensity between the two. The resulting image, the *residual frame*, is coded using a DCT on an 8×8-pixel block basis. As previously discussed, the DCT coefficients are quantised and variable length encoded. Hence, inter frames contain motion information, DCT information, and some video frame header data.

**Fig. 2** Motion prediction in inter frame coding.

### III. IMPACTS OF WIRELESS CHANNEL ERRORS ON VIDEO STREAMS

Wireless channels are typically noisy and transmissions over such channels suffer from a number of channel degradations, such as random bit errors and burst errors due to interference, fading and multipath reflections. The effects of these channel errors, particularly on compressed video, can be very severe [12]. Variable length coding renders video communications more vulnerable to these channel errors. Similarly, predictive coding which highly depends on the previously encoded video frames also makes video transmission quite susceptible to error-prone environments. Both coding algorithms are widely used in video coders which are run in the wireless multimedia communication channels where bandwidth is remarkably limited, and therefore, expensive. However, erroneous variable length coded data causes a loss of synchronisation between the encoder
and the decoder whilst error in predictive coded data causes rapid propagation of channel errors both spatially and temporally through the whole video sequence causing severe quality degradation. Lack of attempts to stop both error effects, either by the encoder or the decoder, ends up with an entire system breakdown.

Mainly, four types of errors can be observed in an erroneous video transmission [13]. The first type causes the “least significant data loss”. For instance, the errors in the wireless communication environment can cause some loss in the texture data of a video frame. This loss might take place only in any MB of the still background scene of the frame, which does not have any motion data at all. Hence, a single bit error on one of the parameters does not have any influence on segments of data other than the affected parameter. In this case, the error is limited to a single MB that does not take part in any further reconstruction process due to the existence of motion on that particular MB. Thus, any kind of temporal error propagation from one frame to another cannot be expected. Similarly, this kind of one background MB error does not cause any loss of synchronisation either resulting in the existence of spatial error propagation within a frame between MBs. A bit error in motion compensated residual pixels in the DCT domain of a MB is an example to this situation, which is not used for prediction. Thus, the damage is localised and confined only to the affected MB without affecting any subsequent data. This kind of error in the video stream is the least destructive one as it does not cause any loss of synchronisation of the decoder with the encoder, however, it still renders the video frame fairly degraded in picture quality.

The second type of error results in “prediction loss”. This type of error is more problematic as it incurs an accumulative damage both in time and space due to prediction. The error propagates to subsequent inter coded frames due to the temporal dependency induced by the motion estimation and compensation processes. This type of error does not
cause any loss of synchronisation either as the decoder is able to flush the right number of
bits of the erroneous motion codewords.

The third type of error gives rise to a “synchronisation loss”. In this case, the
decoder becomes totally unaware of what part of a frame the received information belongs
to. In this circumstance, when the decoder detects an error on a variable length codeword
(VLC), it skips all the forthcoming bits, regardless of their correctness. The skipping
continues until the next error-free re-synchronisation word is detected in the video stream.
Since the decoder discards even all the useful and error-free information it receives between
those two synchronisation points, this process transforms a single bit error corruption into a
bursty channel error.

Due to the nature of the video compression algorithms, the location in the bitstream
where the decoder detects an error might not be the same location where the error has
actually occurred, but some undetermined distance away from it, as followed from Fig. 3.

\[\text{Discarded data}\]

\[\text{Re-synch word} \quad \text{Error location} \quad \text{Error detection} \quad \text{Re-synch word}\]

**Fig. 3** *Error detection in VLC video bitstreams [1].*

Although VLCs provide good compression, they also cause the transmission errors
to propagate along the variable length coded bitstream. Errors are generally detected when
illegal VLCs are found in the bitstream, but in some cases channel errors result in valid
codeword entries in the VLC Huffman table. In these situations, the decoder continues to
decode these erroneous codewords without realising they contain incorrect information. It
takes some time for the decoder to detect the error until it reaches a stage where the
following data is realised to be an illegal VLC. At this point, the decoder has to re-
synchronise itself to the remaining VLC data. This type of error causes one DCT based function to dominate the appearance of the block, which is highly perceivable as a distortion in the image. Another very noticeable type of distortion is caused by incorrectly decoding the DC component of a chrominance block which tends to cause the colour of a 16×16 block to be predominantly pink, cyan or green. Eliminating both of these types of distortion significantly improves the perceived image quality [14].

Finally, the fourth type of error is the worst of all the four kinds as the situation turns out to be more tragic. This takes place when bit errors hit the header data of a video frame. In this case, the decoder can no longer follow the encoder at all. This very disastrous incident might sometimes result in discarding the whole video frame even though the rest of the frame is received correctly.

IV. ROBUST VIDEO TRANSMISSION OVER WIRELESS NETWORKS

Multimedia video compression algorithms should take strict response against the error effects in the wireless transmission media. This response can be achieved by encoding the video data more robust to channel errors. There are several methods that can be exploited:

A. Forward Error Correcting (FEC) Techniques

Video stream can be made more robust to the channel errors by applying FEC techniques, such as Reed-Solomon (RS) codes, convolutional codes, BCH codes [15, 16] and Turbo codes [17], at the encoder. The decoder exploits the use of these codes to detect and correct the bit errors to some extent within the received video stream. FEC techniques are quite effective against short burst and random bit errors, but their performance rapidly degrades with longer duration burst errors. However, FEC codes introduce redundancy to the bitstream due to the reason that they involve the addition of extra parity data to the compressed signal which allows the decoder to correct a certain number of errors whilst
increasing the bit rate. Furthermore, the whole system fails catastrophically whenever a certain worst case is exceeded. Typically, it is much better to apply FEC techniques to provide a certain level of protection to the compressed bitstream, particularly, to the video frame header data, and to let the residual errors be handled by other means of error resilience/concealment algorithms incorporated in video communications.

A.1. Turbo Codes

Turbo codes rely on the Viterbi decoding [18] of received data in a cascaded and iterative fashion. A Turbo encoder consists of two convolutional encoders separated by an $N$-bit interleaver or permuter together with an optional puncturing mechanism. The two convolutional encoders are configured in a manner similar to classical concatenated codes. However, instead of cascading the encoders in a usual serial fashion, they are arranged in a parallel concatenation [19].

In a Turbo decoder, two decoders operating co-operatively and iteratively are generally employed. One decoder provides soft output information to its companion soft-output Viterbi decoder for use in subsequent decoding process. Interleavers and de-interleavers are involved in arranging systematic parity and extrinsic information in the proper sequence for each decoder.

B. Feedback Algorithms for Link Adaptation

Other channel control methods, such as automatic repeat request (ARQ) protocols [20, 21] can be used to lessen error effects, but merely at the expense of reduced channel throughput and unacceptable delays for real-time applications, like video-conferencing. Hence, ARQ protocols are not quite suitable for real-time transmissions. Feedback algorithms offer quite good solutions for wireless applications whereby the video coder adapts itself and regulates its output rate according to the varying channel conditions, only when the return channel
feedback arrives back to the encoder within a short latency. ARQ protocols allow the decoder to request re-transmission of any corrupted data whilst other feedback mechanisms, such as backward channel signalling [22], help the encoder adapt its encoding method to the varying channel conditions. These methods, particularly, play an effective role for coping with packet losses or large burst errors. However, they are less efficient for short duration burst and random bit errors.

C. Zero Redundancy Error Concealment

When an erroneous bitstream is first received by the decoder, the decoder makes an effort to localise the erroneous portion of the video stream. Then, the decoder applies error concealment to minimise the effects of detected errors [23, 24]. There are two major classes of error concealment methods:

**Lost MB recovery:** The first one is the MB concealment whereby entire 8×8-pixel blocks may be missing from the decoded image due to the transmission errors in the channel. The decoder simply replaces the luminance and chrominance of the erroneous MBs with the luminance and chrominance of the corresponding MBs in the previous frame of the video sequence. In other terms, the decoder replaces the erroneous block with the block from the same position in the previous frame. In such case, MB motion vectors are elaborated as zero vectors without any motion data, and hence, resulting in zero residual matrix.

**Lost motion vector recovery:** The same technique can also be applied to the erroneous or lost motion data of the received video frame, and thus, this concealment method can be classified as the second class. Motion vectors of affected MB can be interpolated from neighbouring MBs of the same frame or the same MBs of the previous frames. This technique can also be more effective in case of the use of motion analysis method whereby an estimate of the motion vectors of the lost block can be made using the
information of previous motion data throughout the video sequence. The reason the method is called as zero redundancy is that the encoder literally takes no action against the error-prone channel conditions; on the contrary, it is the decoder which carries out the error concealment.

**D. Error Resilience Tools**

Resilience is a measure of the ability of the video source to adapt itself to varying channel conditions. Resilience might imply some significant changes in the transmission order of video data as well as its structure to render the video stream more robust to wireless channel errors. There are many different error resilience methods and algorithms which have been applied [22]. Principally, error resilience is applied to prevent the error propagation throughout the received video stream. Thus, this prevents a total system breakdown.

**D.1. Error-Resilient Entropy Code (EREC)**

EREC is a method which can be utilised with encoded variable length blocks of data. A variable length block is supposed to be a prefix code, so that the decoder can identify the end of the block it has been decoding without the need for any previous or following block information. The algorithm is based on re-organisation of the variable length blocks in a way that each block starts at a specific position within the code. This particular position is also known to the decoder. Thus, synchronisation is automatically accomplished at the beginning of each block, incurring decreased amount of redundancy compared with FEC algorithms. Fig. 4 depicts an example of the operation of the re-organisation algorithm with six blocks [25].

Channel errors cause the decoder to lose synchronisation with the start and end of blocks. This results in decoding of video data at the wrong location. For some applications, such as speech, this might not be a problem. However, for image and video communications
the effect is the spatially displacement of large portions of the picture with severely degraded QoS [25]. One of the common solutions to this problem is to insert some relatively long unique codewords (synch words) to the encoded video stream at regular intervals, and the other is to encode the entire sequence with FEC techniques which add redundant information as parity bits. However, both techniques add redundancy to the bitstream leading to a lower compression efficiency. Therefore, EREC offers a very efficient way of adding error resilience to the video information with optimum amount of redundancy.

D.2. Reversible Variable Length Codes (RVLC) and Two-way Decoding

Two-way decoding algorithm, with the use of reversible VLCs (RVLCs), presents one of the most effective error resilience techniques, merely with 2–3% bit rate increase due to added overheads. The algorithm of two-way decoding is very simple both in theory and practice. If an error is detected during the decoding process of the received video stream, the decoding is immediately interrupted and the next synch code is searched. Once the next synch code is located, the decoding process is resumed in the reverse direction. Therefore, it is possible to start the decoding process at two different locations in the bitstream, without the need of sending additional side information. To help facilitate two-way decode method,
a set of variable length codes, which can be decoded from both directions (normal mode: from the first bit to the last bit, and reverse mode: from the last bit to the first bit), is used. These modes of operation during the decoding stage imply the right order decoding (normal mode) of the MBs in error-free conditions and the reverse order decoding (reverse mode) of the MBs in error-prone conditions.

Fig. 5a One-way decoding of variable length codes (L: slice length in bits).

Fig. 5b Two-way decoding of variable length codes (L: slice length in bits).

Fig. 5a shows the conventional one-way decoding process. Here, when an error occurrence is detected in the mth MB, the rest of the MBs are skipped, regardless of their correctness, until a new error-free synch word is found. However, as in Fig. 5b, the decoder can re-start decoding backwards as it has reached the next error-free synch word in its two-way decoding mode. The reverse decoding operation ceases when another erroneous MB, such as nth MB in the figure, is detected. Hence, the use of two-way decoding with RVLCs can help salvage some of the invaluable video information from the received data as all the information that can be saved is of high importance for video quality. During both forward and backward decoding, error can be detected in the following cases: either there appears a code which is not listed in the codeword table, or more than 64 coefficients are decoded in a single 8×8-pixel block. There are four patterns for decoding processes according to the
position of errors [22]. In the case that RVLCs are used for error-free environment, it is not necessary to decode it in the backward direction. The combination of RVLCs and the two-way decode is very effective at combating the impacts of both burst and random bit errors.

The method for generating an RVLC is also very simple. For instance, an RVLC can be generated by requiring that a predetermined number of 1s be included in any particular codeword (i.e., if three 1s are included in one code, the RVLC will be 111, 1011, 1101, 11001, 10101, …). Furthermore, 0 can be the first code for this RVLC table. Otherwise, in the case that there are the same number of 0s and 1s in one code, this code can also become an RVLC (for example, 01, 10, 0011, 1100, 001011, 000111, 110100, …) [22]. In this simulation, the exploited RVLC tables were generated as explained in [22, 26].

The next two sections discuss the error control and resilience schemes that were applied and experimented in this paper.

V. SIMULATION OF ERROR CONTROL TECHNIQUES

In this paper, three different error control schemes, namely Turbo codes, error-resilient entropy code (EREC) and two-way decoding in association with reversible variable length codes (RVLC), were individually introduced and experimented. Turbo codes were applied to the administrative data which includes video frame headers and MB control codes, EREC was applied to the motion data and two-way decoding algorithm with RVLCs was applied to the texture data (DCT coefficients) of the video stream in relevant particular simulations. The aim of the experiments was to investigate the performance of each error control algorithm separately in error-prone channel conditions. Besides, the ultimate combination of those three methods was also examined over a simulated wireless channel environment. The error control experiments were carried out with QCIF (176 pixels × 144
lines) size video sequences, and the resilience schemes were applied to the basic H.263 video coding algorithm, which did not have any kind of inherent error resilience.

For the simulation process of the wireless communication environments, both Rayleigh fading and additive white Gaussian noise (AWGN) channels were considered [27, 28]. The effects of both channels on video transmission were taken into consideration as signal-to-noise (S/N) ratio of the video signal. Moreover, S/N factor was determined during the simulations by evaluation of the particular channel characteristics and parameters, such as the frequency bandwidth or impulse response of the channel used. Both channels have deteriorating impacts upon quality of service in wireless communications.

The results and their discussions have been presented in the following section.

VI. RESULTS AND DISCUSSION

A. Turbo Codes Experiment

In the simulations, a Turbo decoder, which consisted of two ½ rate convolutional Viterbi decoders with four iterations and four convolutional interleavers that yielded an overall delay of 936 symbol periods, was used. Due to the additional delay present in the Viterbi decoders, the total delay of the entire Turbo decoder was calculated as 1096 symbol periods. The performance evaluation of Turbo codes was carried out in a Rayleigh fading channel. The objective and subjective results are depicted in Fig. 6 and Fig. 7, respectively.

Fig. 6 presents the simulation results of the 200-frame “carphone” video sequence, encoded at 80 kb/s and 25 fr/s. As seen in Fig. 6, the experiments were run for a Rayleigh fading channel model with two different signal-to-noise (S/N) ratios 4.5 dB and 6.0 dB, respectively. The dotted line represents the PSNR values of the Turbo coded sequence whilst the dashed line represents the PSNR values of the decoded frames when conventional FEC techniques, RS and convolutional codes, are used. The results of the
Turbo codes were obtained for a worse channel condition (S/N = 4.5 dB) than that of the FEC codes (S/N = 6.0 dB). Since ordinary FEC coded sequences produced unacceptable quality below 6 dB and Turbo coded sequences did not suffer much degradation above 5 dB, we have used 6 and 4.5 dB channel S/N ratio conditions for normal FEC and Turbo coded sequences in our experiments, respectively. Therefore, the results are presented here for two different channel conditions bearing two different S/N ratios. It is quite clear from the figure that Turbo codes perform much better than using conventional FEC codes even for worse channel conditions whereby S/N is increased by 1.5 dB. The Turbo coded sequence shows a sharp fall in the PSNR level at a particular frame number due to error accumulation. However, it recovers shortly after this fall. On the other hand, FEC codes present continuous degradation in quality as the number of frames increases. At the 200th frame, the PSNR level difference reaches 14 dB when the PSNR graphs of the FEC codes and Turbo codes are compared.

![Graph](image)

**Fig. 6** The objective results of Turbo codes for “carphone” video sequence.
Fig. 7 presents the subjective results for the same video sequence at the 165th frame. Fig. 7 a shows the effects of AWGN channel errors on original unprotected sequence, Fig. 7 b shows the Rayleigh fading channel effects on RS and ½ rate convolutional coded sequence and finally, Fig. 7 c shows the Rayleigh fading channel effects on Turbo coded sequence. The improvement in the subjective quality with the use of Turbo codes is also quite clear from these figures. The conventional FEC techniques present poorer quality in comparison with the error handling capability of Turbo codes.

![Fig. 7 165th frame of “carphone” sequence, encoded at 80 kb/s and 25 fr/s: a-original unprotected in S/N=7.0 dB AWGN channel, b-RS(255,235) + ½ rate convolutional coded in S/N=6.5 dB Rayleigh fading channel, c-Turbo coded in S/N=5.0 dB Rayleigh fading channel.](image)

**B. Error Resilient Entropy Code (EREC) Experiment**

The EREC experiment was performed with the use of the same video sequence, “carphone”. The sequence was encoded also at 80 kb/s and run at 25 fr/s. The applied error rate corrupted 0.1% of the total bits. Figs. 8 a and b illustrate the subjective results for the 155th frames of the error-prone non-resilient and with EREC video sequences, respectively. The improvement in the decoded picture quality is clearly perceivable.
Fig. 8 155th video frame of “carphone” sequence in 0.1% error rate AWGN channel, encoded at 80 kb/s and 25 fr/s: a-without error resilience, b-with EREC.

C. Reversible Variable Length Codes (RVLC) and Two-way Decoding Experiment

Two-way decoding experiment was also accomplished with the “carphone” video sequence, which was encoded at 28.5 kb/s and 12.5 fr/s. Fig. 9 presents the obtained objective results, PSNR values of three different conditions versus channel BER. The synch words are introduced to each of the slices of a video frame, as seen in Figs. 5a and 5b. A slice is a group of MBs whereby it corresponds to a horizontal row of a QCIF video frame. Since more bits, and therefore, more data is salvaged in the two-way decoding mode, the PSNR improvement can also be clearly noticed from Fig. 9 when the two graphs, one-way decoding and two-way decoding, are compared. In the case that synch words are specially protected, the PSNR values give much better performance than even the usual two-way decoding case. In this figure, it is important to draw the attention to the point that as the BER increases in the channel, the differences between three cases become more discernible in the favour of two-way decoding. Particularly, at the BER=10^{-3} instance, there is roughly 5 dB difference between the worst (one-way decoding) and the best (two-way decoding with protected synch words) cases.
**Fig. 9** PSNR values for “carphone” sequence with two-way decoding and RVLCs.

![Graph showing PSNR values](image)

**Fig. 10** One bit error effect on the 150th frame of “carphone” sequence, encoded at 28.5 kb/s and 12.5 fr/s: a-one-way decoding, b-two-way decoding.

Fig. 10 depicts the 150th video frame subjective results of error-prone one-way decoding and two-way decoding, respectively. In the first image, the error propagation effects are clearly visible as the loss of a slice of the frame. On the other hand, second image shows how the two-way decoding rescued the error-free MBs, by decoding backwards from the end of the erroneous slice or from the start of the new slice, where
indeed a new error-free synch word was detected. Thus, the error was isolated and kept only in one MB, where it originally occurred. This can be observed from the figure as a noticeable square box to the left of the foreground scene.

D. Combination of the Three Algorithms

![Images of four different video frames illustrating the combination of three algorithms: original uncoded, coded error-free, error-prone with FEC techniques, and error-resilient with Turbo codes, EREC, and two-way decoding.]

Fig. 11 124th frame of “suzie” sequence, encoded at 110 kb/s and 25 fr/s: 

- **a**-original uncoded, 
- **b**-coded error-free, 
- **c**-error-prone with FEC techniques (½ rate convolutional codes) only, 
- **d**-error-resilient with ½ rate Turbo codes, EREC and two-way decoding.

As a final simulation, the improvement of the picture quality was elaborated when all the previous three methods were combined. Hence, Fig. 11 illustrates the subjective results obtained from the combination of all the three error control and resilience algorithms implemented on H.263 over an 8 dB Rayleigh fading channel. Fig. 11 shows the 124th frame of “suzie” sequence which was coded at 110 kb/s and 25 fr/s. This final experiment presents the power of combined error resilience technique which comprises Turbo codes applied to the header data, EREC applied to the motion data, and two-way decoding with
RVLCs applied to the texture data (DCT coefficients) of the video stream. The resulting error-resilient image (Fig. 11 d) might not look as good as error-free images (Figs. 11 a and b); however, the picture quality improvement is undoubtedly clear in comparison with the error-prone image (Fig. 11 c).

The experiments were also repeated with different video test sequences and similar conclusive results were obtained from those various tests as presented in this paper.

VII. CONCLUSIONS

In this paper, wireless channel impacts on video communications and the techniques used to alleviate their deteriorating effects on the QoS have been examined. In particular, the three effective error control and resilience algorithms, namely Turbo codes, EREC and two-way decoding with RVLCs were implemented and their performance in error-prone conditions were tested. It has been shown that the errors in noisy wireless channels severely degrade the QoS of video communications. If the referred video communications are, particularly, very low bit rate two-way communications, such as video-phone or video-conferencing, the error effects become intolerable. In this case, it is important to take some radical action against these channel errors and their propagation along the entire video stream. This remedial solution can be achieved by means of FEC techniques, ARQ or feedback algorithms, error concealment methods or error resilience schemes. The most effective way to cope with error effects has been discussed and demonstrated as the error resilience algorithms compared with other techniques since they are applied to stop the error accumulation and to prevent its detrimental effects in video stream with rather less amount of redundancy. Thus complying with this conclusion, Turbo codes have presented around 7.6 dB improvement on average over conventional FEC schemes for 200 frames of the test sequence in same error conditions whilst two-way decoding results have been 5 dB better
than the conventional one-way decoding results for a particular BER of $10^{-3}$. Moreover, the results obtained from the application of all the three error control and resilience algorithms on one video stream proved to give remarkably better objective picture quality than FEC techniques at the same target bit rate.

Acknowledgement

The authors would like to acknowledge Mr P. C. Psiloinis of the Centre for Communication Systems Research for his contribution to this paper.

REFERENCES


based video coders in multimedia communications,” Insights into Mobile Multimedia
Communications, Ed. by D. R. Bull, C. N. Canagarajah and A. R. Nix, pp. 431-443,

and multiplexing layers for very low bit-rate video coding systems,” IEEE Journal on


