Video transmission over mobile satellite systems

S. Dogan*,†, A. H. Sadka and A. M. Kondoz

Centre for Communication Systems Research (CCSR), University of Surrey, Guildford GU2 7XH, Surrey, U.K.

SUMMARY

Mobile satellite communication channels are characterized by long transmission delays, variation of these delays, high bit-error-rates, shadowing and the multipath effect which severely reduce the quality of video services. Error control techniques including feedback mechanisms, error concealment methods, forward error correction techniques and error resilience schemes are examined in this paper for achieving a high-integrity video transmission over a mobile satellite channel. The application of three different error resilience algorithms, namely Turbo codes, error-resilient entropy codes and two-way decoding using reversible codes is presented. Their joint performance is also examined. Furthermore, a low-delay and low-complexity video transcoding algorithm which fully interconnects two very low bit rate video communication standards: MPEG-4 and H.263 is also elaborated. This transcoder works as a gateway tool which links two heterogeneous multimedia networks, such as a mobile satellite network and a land-based network, with negligible processing delay and complexity. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: video coding; mobile satellite systems; error resilience; video transcoding

1. INTRODUCTION

The second half of the 20th century has experienced the rapid deployment and wide spread use of all means of satellite communications, from fixed to broadcast and mobile satellite services. Communications technology has already been prompted for further advances in satellite communications following the introduction of seamless global coverage offered by the latest satellite operators. It has been intended to provide uninterrupted communications services on any part of the earth by the new millennium, and this intention was initiated in the mid-1990s by the international standardization bodies like International Telecommunications Union (ITU), European Telecommunications Standards Institute (ETSI), etc. The infrastructure for this global and universal mobile communications is called Universal Mobile Telecommunications System (UMTS), which is a part of the international mobile communications-2000 (IMT-2000) structure [1].

*Correspondence to: S. Dogan, Centre for Communication Systems Research (CCSR), University of Surrey, Guildford GU2 7XH, Surrey, U.K.
† E-mail: S. Dogan@eim.surrey.ac.uk

Global and highly reliable communications provided by advanced satellite technologies have started to attract very much of the interest in recent years through maritime and aeronautical communications and land-mobile services. Particularly, the noticeably increasing demand on accessing the multimedia information in mobile and remote areas has put mobile satellite communications in a very challenging situation.

Video communications need particular attention due to its high quality of service (QoS) and high bit rate requirements, hence, wide bandwidth demands. A mobile satellite terminal is allowed only to bear a small user antenna having a low gain over temperature ratio (G/T), and economically viable amount of power associated with it for the sake of mobility. Therefore, these constraints have strongly motivated the research on compression of the video data preceding its transmission, as the visual information in raw format is highly redundant [2].

Satellite networks have distinctive features. They provide a broad coverage of spatially dispersed areas and remote locations easily, have broadcasting capability and are easy to install irrespective of geographical constraints. Satellite links can be used as back-up channels for terrestrial networks. On the other hand, they also have intolerable detrimental impacts on the communications quality of service, particularly in video communications. Fixed satellite services have considerably high bit-error-rates (BERs) and significantly long round-trip delays (~ 500 ms for geostationary satellite communications). Moreover, mobile satellite channels are distorted by shadowing, the multipath effect and the delay variations incurred by the propagation effects [3]. The high BER, the shadowing and the multipath effect deteriorate the received video quality severely, and the delay variations which cause jittering distorts the two-way video communications noticeably. Therefore, for reliable communications, the video signal should be made robust against these effects. This can be achieved by error control techniques discussed in this paper. The paper also presents a low-complexity and low-delay video transcoding method intended for the interconnection of two heterogeneous multimedia networks, namely a mobile satellite network and a fixed network. Hence, whilst this procedure provides a seamless interoperability between two different networks with two different operational video standards, MPEG-4 and H.263, it also achieves a high-quality interconnection with low delay which is, particularly, vital in delay effective satellite communications.

The paper is organized as follows. Section 2 gives a brief introduction to video communications. Section 3 discusses the typical mobile satellite channel characteristics whilst Section 4 explains the impacts of these types of channel errors on video streams. Section 5 describes methods for reliable video transmission over such networks. Section 6 presents the error-resilient video experiments, obtained results from these experiments and their discussions. Section 7 introduces a low-delay, low-complexity video transcoding for heterogeneous networks along with the simulation results and their discussion. Finally, Section 8 concludes the paper.

2. VIDEO COMMUNICATIONS OVERVIEW

Video communications imply the transmission of visual information from a source point to a receiver reliably whilst achieving maximum throughput. The communication media have constraints and harmful effects on this transmission process, therefore, the visual information is needed to undergo few processes prior to transmission, such as encoding of the source data by compressing it efficiently, providing it with the ultimate robustness to channel errors, and relaying the encoded robust data over an economically viable power limited channel. The

foremost and the most challenging operation to be exploited is called *video coding* as it is incorporated by video compression which removes the redundancy from the raw video data stream. This is a very important issue to be achieved since bandwidth is a major bottleneck for video communications. *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 Figure 1.

As illustrated in Figure 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 [4]. If frame encoding is performed using only the data from a single frame, an *intra frame* is produced. 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 blocks of 8×8 pixels as followed from Figure 1. After quantization, a DCT block usually contains numerous zero magnitude coefficients, and a few low-frequency values. The coefficients are read in a zig-zag 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 [5], is used to minimize redundancy in the binary output. This process is called as the *entropy coding*. 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
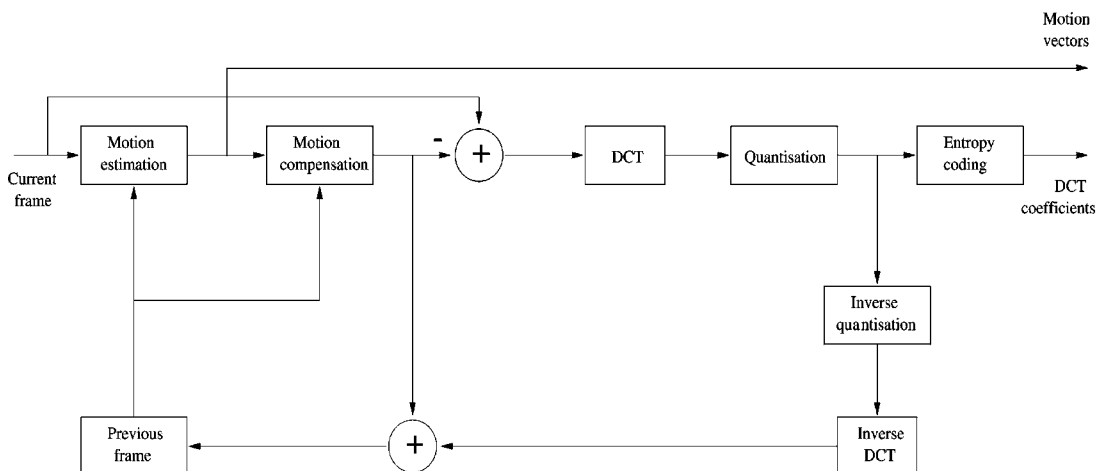

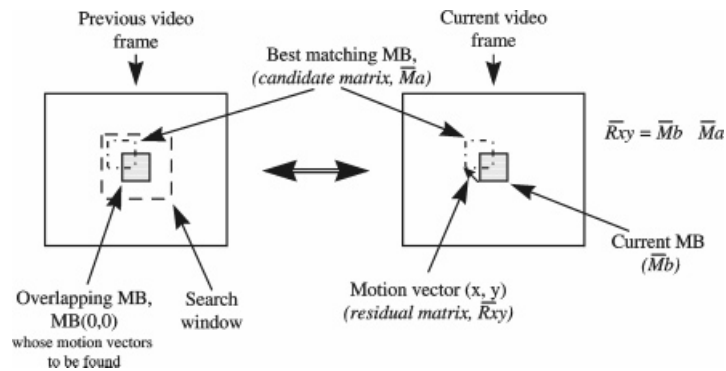Figure 1. A block-based video encoder block diagram [2].

Figure 2. Motion prediction in inter frame coding.

reveal a motion vector corresponding to the relative position of a similar block in the reference frame. This is accomplished within a feedback loop, as seen in Figure 1, which gets the motion information of the previously encoded and locally decoded video frame that has been stored in the previous frame buffer of the encoder. Estimation is performed within a *search window* around the MB as shown in Figure 2. The estimation is carried out by seeking for a best matching block in the previous video frame which coincides with any of the current block whose motion vectors to be found. This search is employed within a search window. When a candidate matrix that corresponds to the best matching block is found, then the motion vectors are calculated with respect to the spatial differences in x and y co-ordinates between this particular block and the current block. The algorithmic notation of this process is presented within Figure 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 quantized and variable length encoded. Hence, inter frames contain motion information, DCT information, and some video frame header data.

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 [6] and H.263 [7] are video-conferencing oriented low and very low bit rate video coders, respectively. MPEG-1 [8] 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 [9] is the standard for generic coding of moving pictures and associated audio at higher bit rates. Finally, MPEG-4 [10] 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 [11].

3. TYPICAL MOBILE SATELLITE CHANNEL CHARACTERISTICS

A fixed satellite service channel is mainly characterized by higher BER due to numerous noise sources and longer propagation delays in comparison with fixed terrestrial communication

networks. The round-trip transmission latency is about 500 ms for geostationary satellite systems whereas the minimum delay can be as long as tens of milliseconds for low earth orbit satellite systems. The high BER is caused by the contribution of different sources of noise along the transmission path and unwanted signals in the form of interference. In fixed satellite communication systems, it is conventionally accepted that an additive white Gaussian noise (AWGN) model provides a generic method for assessing broadband noise which is superimposed to the relayed data through a satellite channel. Hence, an AWGN channel introduces random errors to the conveyed information [12].

A mobile satellite service channel also has the same characteristics as a fixed satellite service channel. Moreover, it also exhibits more problematic features: shadowing and multipath effects. Both effects cause the received signal power and phase to fluctuate in a way that reduces the QoS of satellite communications significantly. When these variations are added up to an inherently high-BER and long-delay transmissions, the received signal becomes no longer acceptable in terms of service quality.

Multipath propagation occurs due to reception of direct satellite signal along with its reflections from surroundings in which a receiver antenna is located as depicted in Figure 3. The additional path length of the reflected satellite signal causes differential delay between the direct and reflected rays. The effect of multipath is to introduce an uncertainty in the instantaneous received signal amplitude and phase. This uncertainty gives rise to variations, and these variations lead to *fades* at time intervals when the received signal power level remains below a threshold. Hence, the overall effect is called *multipath fading* which is a noise-like multiplicative process. The phase and amplitude modulation causes the spectrum of the received signal to broaden [3].

Multipath fading is less dominant for fixed service satellite systems where highly directive antennas are employed. However, it is not the case for mobile satellite service terminals with low to very low gain antennas. Multipath fading is represented by Rician distribution and the probability density function of the instantaneous received power S is given by

$$p_{\text{rice}}(S) = C e^{-C(S+1)} I_0(2C\sqrt{S}) \tag{1}$$

where C is the received carrier-to-multipath ratio (Rice factor) and I_0 is the modified Bessel function of the zeroth order. In the situation where there is no shadowing but only multipath
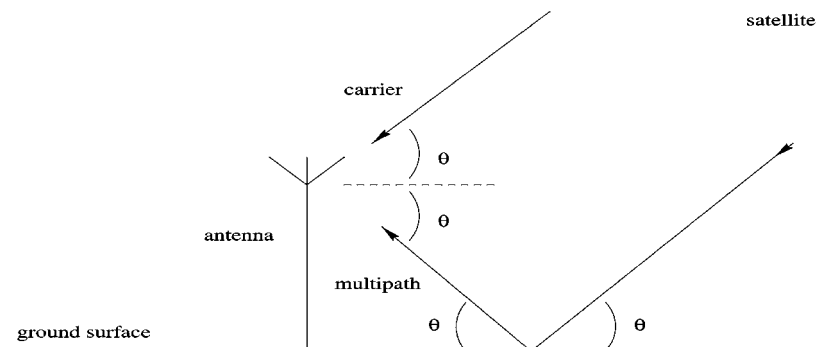

Figure 3. A simple multipath model.

propagation, multipath signal is added onto the direct satellite signal, and hence, the total received signal amplitude forms a Rician process as given in Equation (1) whereby multipath behaviour is modelled by the Rice factor C. If C has larger values, this indicates fewer multipath fades ($C \rightarrow \infty$ for the Gaussian channel) resulting in good transmission characteristics [13].

The fading characteristics highly depend on some criteria, such as the antenna radiation patterns, the elevation angles, the physical characteristics of the reflecting surfaces and the vehicle dynamics.

Multipath is also characterized with delay spread for the reflected path as a result of accumulation of a multiple of reflected elements from the surroundings. This phenomenon results in waste of processing and buffering resources, particularly, in video communications. Hence, its bad impacts are observed on both the quality and the real-time features of video applications. To reduce this effect, more powerful antennas with better reception have to be employed and duplicate video frames have to be detected and discarded within a minimal time limit.

Another very QoS-destructive factor in mobile satellite communications is shadowing. Shadowing happens when the satellite signal is obstructed by buildings, trees, bridges, etc., so that there exists no direct signal path to the mobile terminal. In such a case, the Rice factor C bears smaller values giving rise to severe fading, therefore, the probability distribution reduces to a Rayleigh/log-normal distribution ($C = 0$ for the Rayleigh channel) resulting in a bad channel state. Shadowing mainly dominates at low elevation angles, meanwhile at higher elevation angles both shadowing and multipath effects become less effective [13].

In shadowing, the dominant signal is Rayleigh faded sum of a number of reflected paths. Therefore, a high link power is necessary to reduce the BER to an acceptable range. In unshadowed areas, a BER of 10^{-4} is achievable with moderate link power in spite of the presence of multipath fading [13]. The corruptive effects of multipath fading and shadowing can be alleviated by additional link margins, which are sometimes hard to achieve, particularly in mobile satellite environments where the mobile terminal power is highly restricted. Therefore, the use of various encoding schemes with error resilience mechanisms, interleaving techniques, feedback algorithms or multipath cancellation methods are widely recommended prior to the data transmission.

The power-limited characteristic of satellite communication channels forces the receiver to work at a low signal-to-noise ratio (SNR), and it is up to the receiver to correctly detect the symbols on noisy communication links.

4. VIDEO COMMUNICATIONS OVER MOBILE SATELLITE NETWORKS

Mobile satellite communication 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, multipath reflections and shadowing. The effects of these channel errors, particularly, on compressed video can be very severe [14]. 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 employed over the mobile satellite 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.

Different portions of the encoded visual information in a video frame have different sensitivity characteristics to channel errors. For instance, the motion information of the video data in one frame is much more prone to errors than the texture data of the same frame. Similarly, video headers are accepted to bear the highest importance in a video frame as they are necessary for successful decoding at the receiver. Therefore, a well-known error control technique is the *unequal error protection scheme* where more redundancy is allocated to the most important bits. In this algorithm, the less important bits have a weak error protection or may even be transmitted without any channel coding. In this paper, different error resilience algorithms have been applied to different portions of the video bitstream in accordance with the sensitivity of each data segment to errors.

Mainly, four types of errors can be observed in an incorrectly received video frame [15]. The first type causes a 'data loss' which is rather insignificant compared to the other types. For instance, the errors in the communications environment can cause some loss in the texture data of a video frame. This renders the video frame fairly degraded in picture quality. 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. 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 localized 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 synchronization of the decoder with the encoder.

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 synchronization 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 'synchronization 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-synchronization word is detected in the video stream. Since the decoder discards even all the useful and error-free information it receives between those two synchronization points, this process transforms a single bit error corruption into a bursty channel error.

Owing 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 Figure 4.

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 realizing 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 realized to be an illegal VLC. At this point, the decoder has to re-synchronize itself to the remaining VLC data.
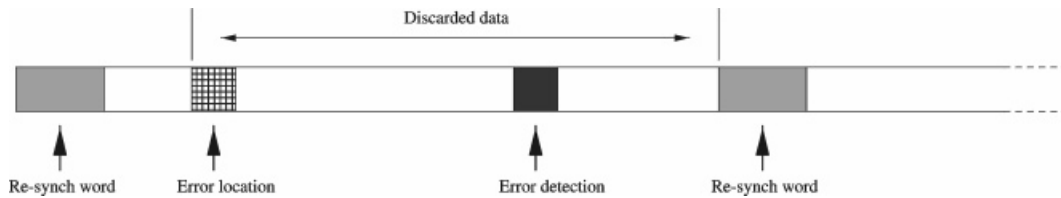
Figure 4. Error detection in VLC video bitstreams [2].

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 [16].

The fourth type of error is the worst of all the four kinds as it comprises header data losses within the video frames due to error conditions. In this case, the decoder can no longer follow the encoder at all. This incident might sometimes result in discarding, and therefore, losing an entire video frame even though the rest of the frame is received correctly.

Finally, the delay in satellite communications also plays an important role on the video communications efficiency and quality. Even though the longer round-trip delay is not classified in the erroneous video transmission category, for satellite applications, this reduces the QoS of two-way video communications significantly since the amount of elapsed time for a feedback signal might well correspond to the duration of few number of video frames.

5. METHODS FOR RELIABLE VIDEO TRANSMISSION

Multimedia video compression algorithms should take strict response against the error effects in the mobile satellite transmission media. This response can be achieved by encoding the video data more robust to channel errors and fades. There are several different methods that can be exploited, such as provision of such a strong link with extra margins or use of automatic repeat request (ARQ) protocols. The former requires excess amount of transmission power to cope with poor channel conditions. However, this forms a very expensive and crude approach to the problem and does not assuage the error effects directly. The latter is also discouraged by longer waiting latency of the return signal in case of a feedback algorithm, and can be employed by the use of a threshold which is a function of both number of errors and round-trip delay. The most suitable resolutions seem to be the utilisation of forward error correction (FEC) techniques and/or error resilience algorithms.

In the following subsections, four major schemes to counteract the deteriorating impacts of error-prone channel conditions on compressed video data transmissions are covered.

5.1. Forward error correction techniques

To render video streams more robust to channel errors, some level of protection must be applied to the compressed video data. The most popular method of protecting digital data is to employ the FEC techniques, such as Reed–Solomon (RS) codes, convolutional codes, BCH codes [17,18]

and Turbo codes [19]. 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 incur some redundancy on the coded data streams due to their nature, and hence, counteract the objective of compression. Using FEC codes is consequently a trade-off between error protection and compression efficiency. Therefore, it is, typically, 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.

5.2. Feedback algorithms for link adaptation

Similarly, 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 utilization and unacceptable delays for real-time applications, like video-conferencing. Feedback algorithms offer quite good solutions for short terrestrial links 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. However, these algorithms are not quite suitable for very long transmission delays like in satellite communications, and therefore, FEC techniques are highly desirable for error detecting and correction purposes in such cases [22]. ARQ protocols allow the decoder to request re-transmission of any corrupted data whilst other feedback mechanisms, such as backward channel signalling [23], 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.

5.3. Zero redundancy error concealment

In this method, 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. The reason this method is called as *zero redundancy* is that the video encoder does not add any redundant information for error protection purposes as error robustness is not accomplished by the encoder. Thus, when an erroneous bitstream is received by the decoder, firstly, the decoder makes an effort to confine the erroneous portion of the video stream. Then, the decoder applies error concealment to minimise the effects of detected errors. When information is lost due to transmission errors, entire 8×8 pixel blocks may be missing from the decoded image. Simply, the decoder 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. The same technique can also be applied to the erroneous or lost motion data of the received video frame. An estimate of the motion vectors of the lost block can be made using the motion vectors of neighbouring or previous blocks.

5.4. 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 communications channel errors. There are many different error resilience methods and algorithms which have been applied [23]. Principally, error resilience is applied to prevent the error propagation throughout the received video stream. Thus, this prevents a total system breakdown.

The following section discusses the error resilience schemes that were applied and experimented in this paper.

6. ERROR-RESILIENT VIDEO COMMUNICATIONS: COMPUTER SIMULATION RESULTS AND DISCUSSIONS

In this paper, three different error resilience 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. The aim of the experiments was to investigate the performance of each error-resilience algorithm separately in error-prone channel conditions. Besides, the ultimate combination of those three methods was also examined over a simulated mobile satellite channel. For the simulation process of the mobile satellite communication environments, both Rayleigh fading and AWGN channels were considered. The error resilience experiments were carried out with quarter common intermediate format (QCIF: 176 pixels $\times$ 144 lines) size video sequences, and the resilience schemes were applied to the basic H.263 video coding algorithm, which does not have any kind of inherent error protection mechanism. Results are presented both subjectively and objectively. The objective results present peak signal-to-noise ratio (PSNR) levels versus video frame numbers whilst the subjective results show particular frames of the test video sequences.

6.1. Turbo codes

The Viterbi algorithm [24] is an efficient technique for decoding powerful convolutional and trellis codes. Turbo codes rely on the Viterbi decoding 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.

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 as seen in Figure 5. Figure 5 depicts the block diagram of an iterative Turbo decoder using maximum *a posteriori* (MAP) [26] decoders. Interleavers and de-interleavers are involved in arranging systematic parity and extrinsic information in the proper sequence for each decoder.

In the simulations, a Turbo decoder, which consisted of two $\frac{1}{2}$ 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, and the objective results are depicted in Figure 6 and the subjective results in Figure 7, respectively.
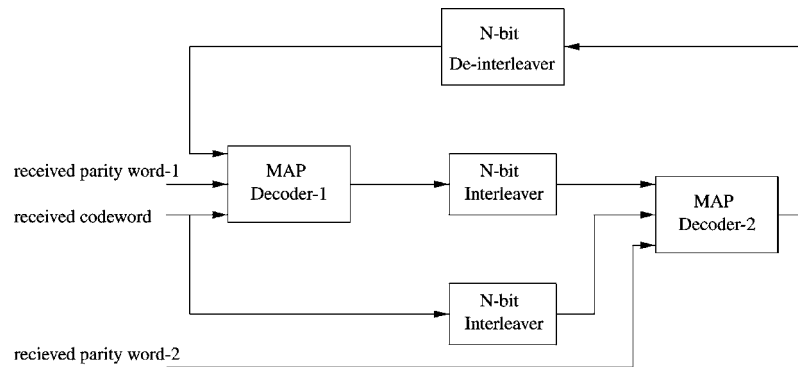
Figure 5. Block diagram of Turbo decoder which uses two MAP decoders operating co-operatively. The inputs to the decoders are received over an AWGN channel. The 'soft' or extrinsic information from decoder-1 is fed into decoder-2. The final decisions may come from either decoder-1 or decoder-2 [25].
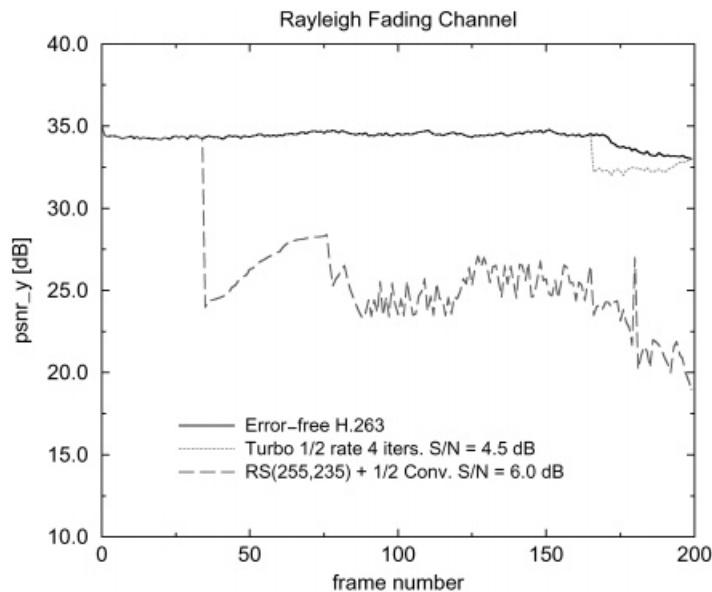

Figure 6. The objective results of Turbo codes for 'carphone' video sequence.

Figure 6 presents the simulation results of the 200-frame 'carphone' video sequence, encoded at 80 kb/s and 25 fr/s. As seen in Figure 6, the experiments were run for a Rayleigh fading channel model with two different signal-to-noise (S/N) ratios 4.5 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

Figure 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) + $\frac{1}{2}$ rate convolutional coded in S/N = 6.5 dB Rayleigh fading channel, (c) Turbo coded in S/N = 5.0 dB Rayleigh fading channel.

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

Figure 7 presents the subjective results for the same video sequence at the 165th frame. Figure 7(a) shows the effects of AWGN channel errors on original unprotected sequence, Figure 7(b) shows the Rayleigh fading channel effects on RS and $\frac{1}{2}$ rate convolutional coded sequence and finally, Figure 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.

6.2. Error-resilient entropy code

EREC is a method which can be utilized with encoded variable length blocks of data. Each 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-organization of the so-called variable length blocks whereby each block starts at a specific position within the code. This particular position is also known to the decoder. Thus, synchronization is automatically accomplished at the beginning of each block. Figure 8 depicts a simple example of the operation of the re-organization algorithm with six blocks of lengths 11, 9, 4, 3, 9, 6, placed in 6 equal length slots of 7 bits [27].

Channel errors cause the decoder to lose synchronization 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 depending on its error resilience to errors which differs from one speech codec to another. However, for image and video communications the effect is the spatially displaced of large portions of the picture with severely degraded QoS [27]. One of the
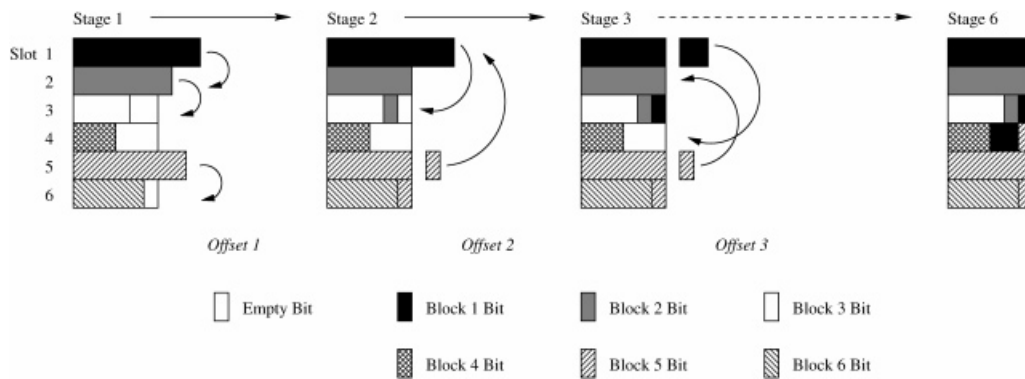
Figure 8. The EREC algorithm [27].

Figure 9. 155th video frame of 'carphone' sequence in 0.1 per cent error rate AWGN channel, encoded at 80 kb/s and 25 fr/s: (a) without error resilience, (b) with EREC.

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. 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 without including any redundancy in the data stream.

The EREC experiment was performed with the use of same video sequence, 'carphone'. The sequence was encoded also at 80 kb/s and run at 25 fr/s. The applied error rate corrupted 0.1per cent of the total bits. Figures 9(a) and 9(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.

6.3. 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 per cent bit rate increase due to added overhead. 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.

During forward and/or backward decoding, errors can be detected in the following cases: either there appears an unacceptable codeword which is not listed in the Huffman table as explained in Section 4, or more than 64 coefficients are decoded in a single 8×8 pixel block. It might happen that an error occurrence results in a new admissible codeword but the decoder will be able to detect the error when it loses the synchronisation after a certain number of bits. Thereby, the probability of having an undetectable error is very unlikely. In the case that RVLCs are used for error-free environment, it is not necessary to decode in the backward direction. The combination of RVLCs and the two-way decode is very effective at resolving the impacts of both burst and random bit errors.

Figure 10(a) 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 Figure 10(b), 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.
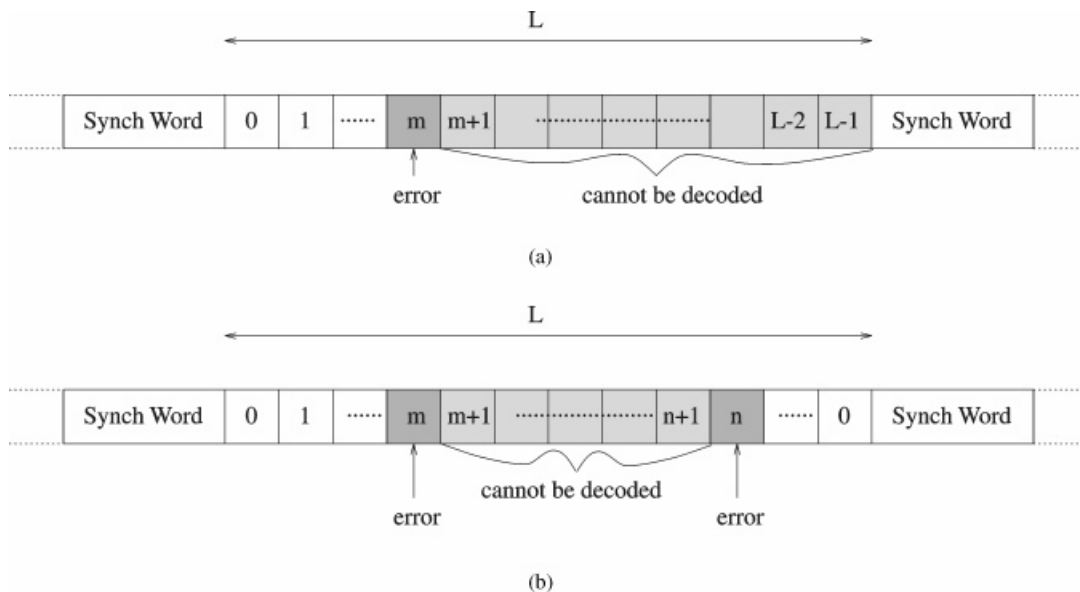

Figure 10. (a) One-way decoding of variable length codes (L: slice length in bits). (b) Two-way decoding of variable length codes (L: slice length in bits).
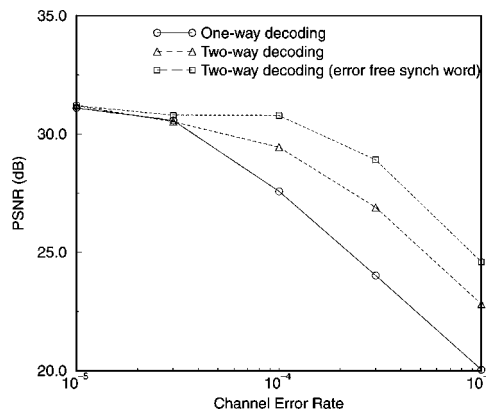
Figure 11. PSNR values for 'carphone' sequence with two-way decoding and RVLCs.

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. Figure 11 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 Figures 10(a) and 10(b). A *slice* is a group of MBs whereby it corresponds to a horizontal row of a 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 Figure 11 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 are much higher than the usual two-way decoding case giving better performance results under error conditions. 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.

Figure 12 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 object.

Finally, Figure 13 illustrates the subjective results obtained from the combination of all the three error resilience algorithms implemented on H.263 over an 8 dB Rayleigh fading channel. Figure 13 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 administrative data which includes video frame header data and MB control codes, 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 (Figure 13(d)) might not look as good as error-free images (Figure 13(a) and (b)); however, the picture quality improvement is undoubtedly clear in comparison with the error-prone image (Figure 13(c)).

Figure 12. 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.
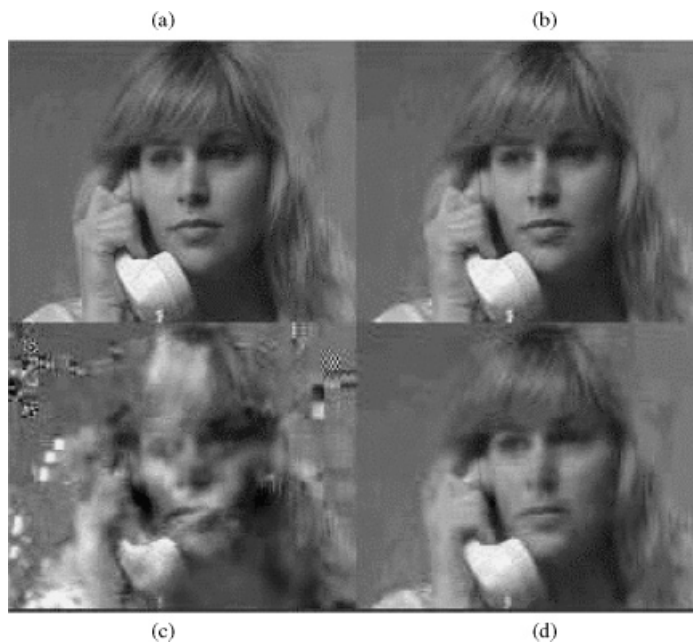

Figure 13. 124th frame of 'suzie' sequence, encoded at 110 kb/s and 25 fr/s: (a) original uncoded, (b) coded error-free, (c) error-prone with FEC techniques ($\frac{1}{2}$ rate convolutional codes) only, (d) error-resilient with $\frac{1}{2}$ rate Turbo codes, EREC and two-way decoding.

7. LOW-DELAY LOW-COMPLEXITY VIDEO TRANSCODING

This section presents an evaluation of a proposed solution to the interoperability problem which is believed to be experienced in the near future when two different multimedia networks will need to communicate with each other. The prospective problem is that a multimedia application running on a mobile satellite communications environment requires an efficient linkage to

a similar or diverse type of application operating on a fixed network. This paper handles the problem from the video communications point of view. Hence, a video transcoder is introduced to employ necessary translations between two video data stream syntax to achieve a low-complexity and low-delay interconnection. This achievement provides a full-scale end-to-end support of diverse types of video coding standards whilst reducing the necessary processing time as the operation does not involve time consuming conventional decode and re-encode processes. It is indeed a very significant improvement as the mobile satellite networks have already inherent delay characteristics.

7.1. Video transcoding

Video transcoding is essential for end-to-end compatibility of two different and heterogeneous networks operating with different characteristics and constraints. This is an interoperability issue considered as a necessity for global seamless interconnection of different network topologies. Video transcoding is performed on the compressed bitstream domain. The conversion of a compressed bitstream format into another compressed format is called *transcoding* and the tool that performs this conversion is called a *transcoder*. The conversion is needed to convey an encoded bitstream from one algorithm to another or from one transmission channel to another with different constraints and characteristics. Hence, two major distinct types of transcoding schemes arise: algorithm (standard) conversions and rate or resolution conversions in the same algorithm. The former provides a coded bitstream of one standard to accommodate onto a different standard without performing cascaded decoding/re-encoding whilst the latter employs bit rate reduction, resolution reduction or frame-rate reduction according to the varying channel characteristics [28]. Successful implementations of the latter type of transcoders in pixel domain [29] as well as in frequency domain [30] have been presented. In this type of transcoding, noise and artefacts which are due to drifting of pictures that increases with increasing number of predicted frames are the main problems. Increasing distortion caused by the *drift* occurs due to the mismatch of reconstructed pictures in the encoder and the decoder [31].

The transcoding mechanism elaborated in this work is an algorithm-conversion type of transcoding that employs a tandem of two video coding standards operating at very low bit rates: H.263 and MPEG-4. *Tandem* is the linking or interconnection of two standards to each other. Transcoding is a method of tandem in which translation of the syntax of two standards takes place without any further decoding and re-encoding processes. Therefore, the main aim of transcoding is to optimise the processing power and the processing time, and to improve the QoS by eliminating the distortion caused by cascaded decoding and re-encoding processes which exist at the interconnection point of two different video standards. This interconnection point is the *video gateway* as seen in Figure 14. Since the video transcoder only achieves a straightforward translation and mapping operation of the syntax of two video coders, it does not involve in very complex and time-consuming motion estimation and compensation or DCT/inverse DCT (IDCT) processes. Thus, the computational complexity discernibly reduces giving rise to much shorter processing delays which is very desirable in long-delay satellite networks. Therefore, the computational complexity is less than a single decode operation, and also much less than encoding process whereby motion estimation, compensation and DCT/IDCT operations consume much of the processing power and time. In the next subsection, simulation results are presented for a video transcoder which fully links the two video standards, MPEG-4 and H.263, operating at very low bit rates and on different networks: MPEG-4 on a mobile satellite network
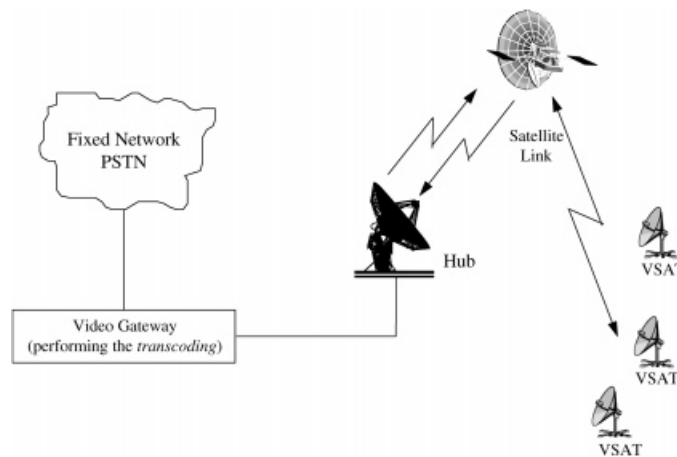
Figure 14. The interconnection of two heterogeneous multimedia networks via transcoding.

and H.263 on a circuit-switched network. This achievement is the extension of our previous research where we implemented a novel bi-directional MPEG-4/H.263 video transcoder for interoperability between heterogeneous multimedia networks [32].

7.2. Results and discussion

The transcoding simulations were carried out by using two sets of video sequences: 150-frame 'suzie' sequence and 200-frame 'foreman' sequence. The results are obtained and presented for both directions, from MPEG-4 to H.263 and vice versa, and also for three different simulations: (i) direct encoding/decoding, (ii) transcoding and (iii) cascaded decoding/re-encoding. Figure 15 depicts the objective results for both sequences and Figure 16 shows the subjective results for 'foreman' sequence only.

   Figure 15(a) and (c) illustrates PSNR variations for the MPEG-4 to H.263 direction whilst Figure 15(b) and (d) presents the reverse direction PSNRs. In all four graphs, it is clearly seen that transcoding results give a performance at least as good as direct encoding/decoding simulation results. Conversely, cascaded decoding/re-encoding PSNRs are on average 1–1.5 dB less than transcoding PSNRs. An average of 0.5 dB difference between two sets of results is caused by the usage of four motion vectors per MB for MPEG-4 to H.263 direction which increases the overall picture quality, particularly in high motion areas. This is clearly visible by examining Figure 15(a)–(b) and (c)–(d) pairs.

   The subjective results of 200th frame of 'foreman' sequence can be seen in Figure 16. The top row depicts MPEG-4 to H.263 direction whilst the bottom row shows the H.263 to MPEG-4 direction for direct encoding/decoding, transcoding and cascaded decoding/re-encoding results, respectively. Both figures (c) and (f) present some degraded quality in comparison with the others due to the distortion caused by the excessive quantisation losses and the re-evaluation of the previously estimated and compensated motion vectors of the decoding and re-encoding processes.
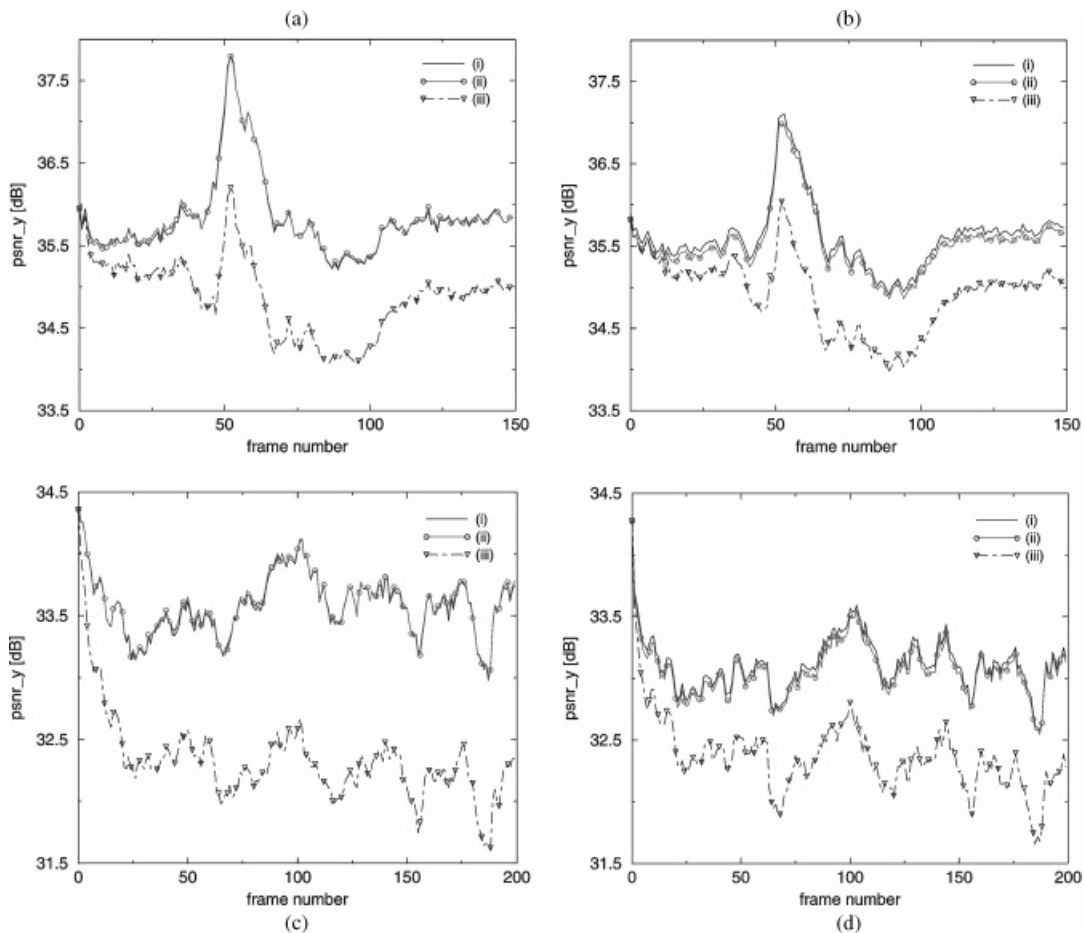
Figure 15. PSNR variations with and without transcoding: (a) and (b) 'suzie' sequence, (c) and (d) 'foreman' sequence; (i) direct encoding/decoding, (ii) transcoding, (iii) cascaded decoding/re-encoding.

8. CONCLUSIONS

This paper has discussed the effects of mobile satellite channels on video communications. It has been shown that high error rates severely degrade the perceptual video quality. Error robust video communication has been discussed, and different remedial solutions, such as FEC techniques and error-resilience schemes, have been presented. In addition to these, the performances of three error-resilience algorithms, namely Turbo codes, EREC and two-way decoding with RVLCs, have also been demonstrated. It has been shown that video sequences encoded using error-resilience tools give far superior subjective and objective quality measures compared to equivalent non-error-resilient streams when the bitsreams are subjected to bit errors. As far as the obtained simulation results are concerned, Turbo codes have presented approximately ~ 7.6 dB improvement on average over conventional FEC schemes for 200 frames of the experimented test sequence in same error conditions. Meanwhile, two-way decoding results have been about 5 dB
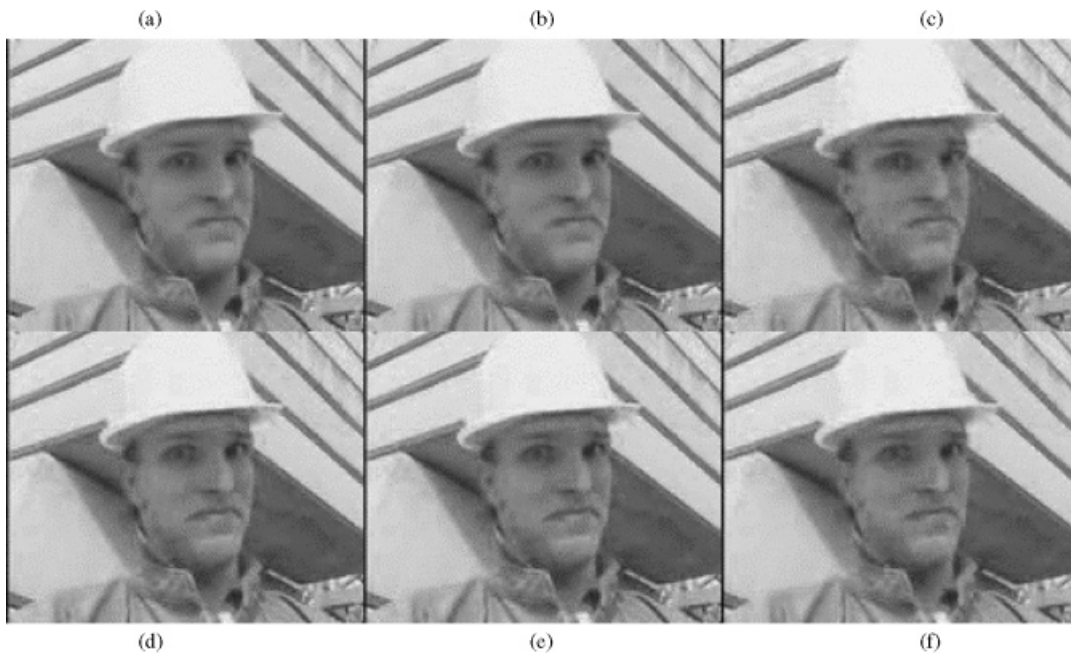
Figure 16. Subjective results for the 200th frame of the 'foreman' sequence, encoded at 100 kb/s on average and 25 fr/s: *top row*-MPEG-4 to H.263 path, *bottom row*-H.263 to MPEG-4 path; (a) and (d) direct encoding/decoding, (b) and (e) transcoding, (c) and (f) cascaded decoding/re-encoding.

better than the conventional one-way decoding results for a particular BER of 10^{-3}. The combination of the studied error-resilience algorithms has given a noticeably better objective picture quality than FEC techniques at the same target bit rate. Moreover, a low-delay, low-complexity video transcoder that accomplishes a bi-directional syntax translation between MPEG-4 and H.263 video standards has been also presented. It has been shown that the transcoder can be used to link two or more heterogeneous networks, for instance a fixed network and a mobile satellite network, and gives a better performance in terms of picture quality than conventional video gateways which implement cascaded decoding and re-encoding processes. During the simulations, the achieved picture quality improvement has been around 1–1.5 dB on average. Furthermore, this transcoder achieves a better performance without any significant processing delay that is imposed by the conventional tandem operation.

REFERENCES

1. Dahlman E, Gudmundson B, Nilsson M, Skold J. UMTS/IMT-2000 based on wideband CDMA. *IEEE Communications Magazine* 1998; **36**(9):70–80.

2. Talluri R. Error-resilient video coding in the ISO MPEG-4 standard. *IEEE Communications Magazine* 1998; **36**(6):112–119.
3. Evans BG (ed.). *Satellite Communication Systems* (2nd edn). Peter Peregrinus Ltd: Exeter, England, 1991.
4. Rao KR, Hwang JJ. *Techniques and Standards for Image, Video and Audio Coding*. Prentice-Hall: U.S.A., 1996.
5. Tekalp AM. *Digital Video Processing*. Prentice-Hall: U.S.A., 1995.
6. CCITT Recommendation H.261. Video codec for audio visual services at p × 64 kbits/s. COM XV-R 37-E, 1990.
7. Draft ITU-T Recommendation H.263. Video coding for low bitrate communication. May 1996.
8. ISO/IEC CD 11172. Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbits/s. December 1991.
9. ISO/IEC CD 13818-2. Generic coding of moving pictures and associated audio. November 1993.
10. ISO/IEC JTC1/SC29/WG11 MPEG97/N1796. Video verification model version 8.0. In *Stockholm Meeting*, Ebrahimi T (ed.), July 1997.
11. Reader C. MPEG-4: coding for content, interactivity and universal accessibility. *Optical Engineering* 1996; **35**(1):104–108.
12. Maral G, Bousquet M. *Satellite Communications Systems* (2nd edn). Wiley: Chichester, England, 1993.
13. Lutz E, Cygan D, Dippold M, Dolainsky F, Papke W. The land mobile satellite communication channel—recording, statistics, and channel model. *IEEE Transactions on Vehicular Technology* 1991; **40**(2):375–385.
14. MacDonald N. Transmission of compressed video over radio links. *BT Technology Journal* 1993; **11**(2):182–185.
15. Sadka AH, Eryurtlu F, Kondoz AM. Aspects of error resilience for block-based video coders in multimedia communications. In *Insights into Mobile Multimedia Communications*, Bull D, Canagarajah N, Nix A (eds). Academic Press: Great Britain, 1999; 431–443.
16. Lee WS, Pickering MR, Frater MR, Arnold JF. Error resilience in video and multiplexing layers for very low bit-rate video coding systems. *IEEE Journal on Selected Areas in Communications*, 1997; **15**(9):1764–1774.
17. Wicker SB. *Error Control Systems for Digital Communication and Storage*. Prentice-Hall: Englewood Cliffs, NJ, U.S.A., 1995.
18. Sweeney P. *Error Control Coding: An Introduction*. Prentice-Hall: Wiltshire, Great Britain, 1991.
19. Berrou C, Glavieux A, Thitimajshima P. Near Shannon limit error-correcting coding and decoding: turbo-codes. In *Conference on Proceedings IEEE ICC'93*, Geneva, May 1993; 1064–1070.
20. Halsall F. *Data Communications, Computer Networks and Open Systems* (4th edn). Addison-Wesley Publishing Company Inc.: U.S.A., 1996.
21. Chang LF. Throughput estimation of ARQ protocols for a Rayleigh fading channel using fade-duration and interfade-duration statistics. *IEEE Transactions on Vehicular Technology*, part 2 1991; **40**(1):223–229.
22. Dunlop J, Smith DG. *Telecommunications Engineering* (3rd edn). Chapman & Hall: London, Great Britain, 1994.
23. ISO/IEC JTC/SC29/WG11 MPEG97/N1646. 'Description of error resilient core experiments. *Bristol Meeting*, April 1997.
24. Forney Jr GD. The Viterbi algorithm. *Proceedings of the IEEE* 1973; **61**:268–278.
25. Ryan WE. A Turbo code tutorial. http://www.ee.virginia.edu/CSL/turbo_codes /overview.html
26. Bahl L, Cocke J, Jelinek F, Raviv J. Optimal decoding of linear codes for minimising symbol error rate. *IEEE Transactions on Information Theory* 1974; **20**:284–287.
27. Redmill DW, Kingsbury NG. The EREC: an error-resilient technique for coding variable-length blocks of data. *IEEE Transactions on Image Processing* 1996; **5**(4):565–574.
28. Bjork N, Christopoulos C. Transcoder architectures for video coding. *IEEE Transactions on Consumer Electronics* 1998; **44**(1):88–98.
29. Morrison G. Video transcoders with low delay. *IEICE Transactions on Communications* 1997; **E80-B**(6):963–969.
30. Assuncao PAA, Ghanbari M. Transcoding of MPEG-2 video in the frequency domain. In *IEEE International Conference in Acoustics, Speech and Signal Processing, ICASSP'97*, Munich-Germany, vol. 4. April 1997; 2633–2636.
31. Youn J, Sun MT, Lin CW. Motion estimation for high performance transcoding. *IEEE Transactions on Consumer Electronics* 1998; **44**(3):649–658.
32. Dogan S, Sadka AH, Kondoz AM. Efficient MPEG-4/H.263 video transcoder for interoperability between heterogeneous multimedia networks. *IEE Electronics Letters* 1999; **35**(11):863–864.