A LOW-COMPLEXITY ERROR-RESILIENT H.263 CODER

Masoud Khansari and Vasudev Bhaskaran

Hewlett-Packard Laboratories, Palo Alto, California, 94304

ABSTRACT

We propose a low-complexity error resilience method which is compatible with the emerging H.263 standard. The proposed method, which corrects one slice of lost information exactly or partially, is specifically suited to the packet networks such as internet where the majority of the errors are due to packet losses. The proposed method is simple and to decode the overhead information no extra software or hardware is required. The amount of generated overhead is minimal as is the delay to reconstruct lost information. Using this method, it is possible to easily trade-off bits between the source and protection data, hence achieving a simple adaptation method to the current status of the channel.

1. INTRODUCTION

Video compression algorithms such as H.261 or H.263 can compress motion sequences efficiently. However, the compressed information is more sensitive to transmission errors caused by channel impairments. In fact, more compressed the signal, more information is carried by an individual bit resulting in increased distortion in the reconstructed signal if that bit is not correctly decoded. Moreover, catastrophic events such as loss of synchronization can occur by incorrectly decoding a single bit.

Recent applications such as video conferencing over the internet or video transmission over wireless links requires low bit-rate coding. Sophisticated methods are used to remove signal redundancies in both spatial and temporal dimensions. However, the transmission lins are not clean. Furthermore, the transmission channels are not clean. For example, one can expect Bit Error Rate (BER) as high as 10^{-3} in the wireless links. In the case of internet it is not unusual to lose about 10% of the transmitted packets. Therefore the transmission of highly compressed signals over channels with high error rate is quite challenging. The fact that errors can occur in burst (e.g. when the radio link is in fade) adds another challenging dimension to this problem.

Another aspect of the problem is the fact that video transmission is usually a packet-based stream. In such an environment it is possible to lose an entire packet. This can for example be due to buffer overflow in the routers or switches in the intermediate nodes. Usually, these packets contain a powerful enough Cyclic Redundancy Codes (CRC) or check-sum to trap erroneously received packets. In such cases, it is still possible to try to parse and decode the packet as long as possible but the decoded information is not reliable and should be used with care. Added to the above, for real-time application with stringent delay constraints, packets arriving late at the destination should be considered as loss packets. It is therefore necessary to issues surrounding packetization since data organization and arrangement can significantly affect the robustness of the video transmission system. It is then of no surprise to see several standardization bodies such as ITU (H.223), ISO-MPEG (MPEG-4) and IETF (e.g. drafts proposing RTP payload format for encapsulating real-time bit streams) addressing this issue. Ideally, the goal is to be able to decode each packet independently from the other packets so that the loss of one packet does not affect the decoding of the other packets. In other words, each packet should be an *independent* and self-contained unit.

An important aspect of these new transmission media is their time-varying characteristics. For example, congestion can occur in the internet causing long transmission delays. A simple remedy for this problem is to make the sources lower their transmission rate, hence permitting the network to clear the backlog. In the case of wireless radio link, during a fade the BER can be as high as one half resulting in practically no transmission of information during that period. It is therefore highly desirable to have a flexible source coder that can readily adjust to the current status of the channel. For example, by being able to adjust the amount of the protection one can avoid unnecessary overhead when the channel is clean, thus improving the overall performance of the system considerably.

2. PACKETIZATION OF H.263 DATA STREAM

H.263 has emerged as the dominant video coding standard for applications requiring low bit-rate coding, replacing H.261 which was mainly intended for video conferencing applications. It is a hybrid motion-compensated coder based on half pixel accuracy motion estimation where each motion vector is encoded differentially using a prediction window of the three motion vectors of the surrounding macro-blocks. This is in contrast to H.261 where the motion estimation uses full pixel accuracy and the motion vector of the previous macro-block (MB) is used as the prediction for encoding the motion vector of the current MB.

Similar to H.261, each group of block (GOB) has its own header, but unlike H.261 the position of the GOB header is not fixed and can be varied to contain one or more slices – each slice being one horizontal row of MBs. Note that the motion vectors of the first slice of each GOB are effectively encoded in a similar fashion as H.261 using the adjacent MB as the predictor. As a result, if we limit each GOB to consist of only one slice, then no information from the pre-

vious slice is needed to decode this GOB. A single packet can then be used to packetize this GOB, and it is not necessary to pad any additional information from the previous slices to this packet header. This is in fact proposed as one of the transmission modes (Mode A) for the RTP payload format of H.263 video stream [1]. Modes B and C allow for fragmentation at the MB boundaries but require considerably more overhead (two and three time more, respectively) which can be prohibitive at low bit-rate regimes. Mode A also provides an easy error recovery method since the picture and the GOB header can be easily identified at the beginning of each packet payload. The main disadvantage of Mode A is its inflexibility with respect to the network packet size – the bits generated for each GOB should be smaller than the packet size. This, however, can in most circumstances be overcome by using a proper rate allocation mechanism.

3. ERROR RECOVERY METHODS

Error recovery methods fall into two general categories of open- and closed-loop methods. In the closed-loop methods, a back channel from the receiver to the transmitter is maintained. This back channel conveys the status of the transmitted packets to the transmitter, providing it with the possibility of either retransmitting the erroneously received packets or containing the effect of their losses [2, 3]. The main drawback of the packet retransmission is the added delay which can be prohibitive for real-time applications. When this delay is tolerable, the closed loop method improves the overall performance of the system considerably and should be utilized. If it is not possible to have a back channel (e.g. broadcast channel) or the added delay or complexity is prohibitive, then the open-loop error recovery method is used.

In open-loop methods such as Forward Error Correction (FEC) or error concealment, the recovery from packet errors is the responsibility of the receiver. In the case of FEC, the transmitter adds redundant parity bits which can be used, to an extent, to recover lost information. Since when a packet is lost, the entire GOB is lost, the parity bits should be packetized separately. Error concealment tries to reconstruct the lost data using the information available in the same or previously decoded frames. A popular scheme is to replace the lost GOB by the GOB from the previous picture frame at the same spatial location. This simple method is effective in most situations except when there is a high amount of motion and activity in the scene. Also for a hybrid coder where temporal prediction is used, the reconstructed sequence at the transmitter is not exactly the same as that at the transmitter. This results in error propagation along the temporal dimension and is known as the drift problem. As we will show that for coders such as H.261 or H.263, this error propagation tends to be persistent.

The method proposed in this paper is an open-loop mechanism which tries to remedy the above deficiency. Similar to FEC, redundant information is added but unlike FEC, this is done during the compression stage and not after. This provides the interesting possibility of being able to adaptively change (with high enough granularity) the number of bits allocated to the parity information which, at the least, is a difficult task using FEC.

4. PROPOSED METHOD

Since each slice is packetized independently, a loss of one packet can result in a loss of at most one slice of information. We now propose a recovery method for the decoder, which can recover (exactly or partially) this loss of information. This is done through construction of a new slice which we call *erasure* slice (see Figure 1). This erasure slice is coded using standard H.263 and hence the generated bit stream is a *valid* stream that can be decoded using the same decoder (either in hardware or software).

If the amount of motion is limited within the scene, then copying the lost GOB from the previous picture is satisfactory. It is therefore necessary to have a mechanism to discover pictures with high activity. In our work we use the the sum of motion vectors within a frame, i.e. if $MV_x(i,j)$ and $MV_y(i,j)$ are the x and y motion vector components of the jth MB of the ith slice then we define activity parameter [1] as

$$\sum_i \sum_j (|MV_x(i,j)| + |MV_y(i,j)|). \qquad (1)$$

Redundant information is only transmitted when this parameter is greater than a predefined threshold $ACTH$. Note that $ACTH$ can be changed dynamically.

Information in each MB can be classified into two categories. The first is the information specifying the parameters used in decoding the residual information (DCT coefficients) and the second is the residual information itself. Examples of the first category of the information are motion vectors, quantization parameters and the macro-block type. Any loss of information in this category may have catastrophic consequences and special care is needed to protect this information. Our strategy is, for a lost MB, to be able to reconstruct the first category losslessly whereas the residual information can be reconstructed with loss based on the available bit budget.

Each MB of the erasure slice is constructed using MBs of the other slices located in the same column. For example the third MB of the erasure slice is constructed using only the third MBs of the previous slices - Figure 1. Specifically the residual information is found by summing the residual information of the MBs of the same column. Note that this sum can be taken either after or before quantization - we use the quantization output in our simulation. The resultant coefficients are then divided by a factor which is known at both the receiver and the transmitter. This is done to limit the dynamic range of the coefficients resulting in a lower bit rate (less ESCAPE mode is used at the encoder). In our simulation we found that the factor 2 provides a good tradeoff. Furthermore, hard tresholding is used – any coefficient whose absolute value is less than T is set to zero. This causes longer runs of zeros and hence lowers the number of the bits generated by the erasure slice. We use the value T and also the activity threshold $ACTH$ to control the bit rate

[1] We also considered energy of the residual signal. This can however be misleading in the low-rate regime when coarse quantization is used.

generated by the erasure slice. When at most one of the packets containing information of the picture is lost then it is possible to partially reconstruct the lost information using this erasure slice (the reconstruction is exact if $T = 0$ and the division factor is set to 1).

As was stated previously, any loss of information in the motion estimation information can be catastrophic. We therefore propose the following method to generate the motion vector field of the erasure slice. Each component of the motion vector ($MV_x(i,j)$ and $MV_y(i,j)$) is in the range $[-16, 15.5]$. Then $MVE_x(i)$ (the x component of ith MB of the erasure slice) is found as

$$MVE_x(i) = \sum_j MV_x(i,j),$$

where the summation is taken as modulo summation to ensure the value of the summation is in the range $[-16, 15.5]$. The value of $MVE_y(i)$ is found similarly. It is therefore possible to encode $MVE_x(i)$ and $MVE_y(i)$ using the same VLC table for MVD provided by H.263 standard. Note that if the ith slice of the picture is lost, all the motion vectors of the MBs of this slice can be reconstructed losslessly using the information available in the erasure slice.

The value of DQUANT of each macro-block of the erasure slice is found in a similar lossless fashion. The DQUANT parameter is in the range $[-2, 2]$ and modulo summation is used to ensure that the value always remains within this range. Note that if the MB type of any of the MBs in the same column is INTER+Q (i.e. the quantization parameter is modified through DQUANT) then we set the type of the corresponding MB in the erasure slice to INTER+Q. In other words, the type of a MB in the erasure slice is INTER only if all the MBs in that column are of INTER type. We also restrict the use of INTRA MBs to intra-frame pictures.

5. THE ALGORITHM PERFORMANCE

We used the "carphone" sequence to test our proposed recovery algorithm. The sequence is in QCIF format and the frame rate is 10 frames/sec. We assumed that the transmission rate is 28 Kbit/sec and encoded the sequence at this fixed rate. The activity threshold parameter ($ACTH$) was set to 300 which roughly translates to 1.5 pixel for each component of the motion vector (on the average). We also set the value of $T = 2$ (i.e. in the erasure slice all the DCT coefficients with absolute value smaller than 3 are set to zero). These values were found empirically and provided the desired trade-off and were kept constant throughout the simulation even though it is possible to dynamically vary them.

Figure 2 shows the number of bits generated by our method where the erasure slice bit sequence is shown separately. The bit sequence of the baseline coder is also shown for comparison. [2]. The erasure slice is constructed only for those frames that have activity parameters greater than 300 and in those cases the amount of overhead generated is between 10% and 15% of the total bit budget. Figure 3 shows the PSNR performance of our method and the baseline coder when the fifth GOB of frame number 10 is lost. The concealment method used for the baseline coder is to replace the lost GOB by the GOB at the same spatial location in the previous frame. Note that for the frame with the lost GOB, the PSNR performance of our method is better than that of the baseline coder by about 1.5 dB and continues to do so for a relatively long period of time in the future (about 1.5 seconds or 15 frames). This shows that for motion-compensated based hybrid coders (such as H.263), the errors due to transmission losses tend to be persistent, so simple, effective error concealment methods are needed. Figure 4 shows the reconstructed and the residual error pictures where both our method and the baseline method are shown. Figure 5 shows the reconstructed pictures after 1 second (frame number 20). As it is clear from the pictures, our method has corrected the effect of GOB loss, while the baseline coder continues to suffer from the loss of the GOB.

[2]By baseline coder we mean H.263 coder running at 28 Kbit/sec.

6. SUMMARY AND CONCLUSIONS

We proposed a loss recovery method which is most suited to packet networks such as the internet. This method has the following strengths: A generic H.263 decoder can be used to decode the overhead information, i.e. there is no need for additional hardware or software. The overhead information are packetized separately without any need to modify the transport layer. The overhead packet looks exactly the same as every other packet. Also the overhead information can be ignored by simply dropping the overhead packets and if this information is not needed one can simply ask the transmitter to stop transmitting these packets. The delay incurred in reconstructing the lost MBs is only the time it takes to decode the erasure slice. The overhead due to the erasure slice is minimal, i.e. less than 10%, and this overhead is sufficient to recover one erroneous slice. The reconstruction of the lost MBs can be either lossy or lossless with the quality of the reconstruction being dependent on the rate allocated to the erasure slice.

7. REFERENCES

[1] RTP Payload Format for H.263 Video Stream, Internet Engineering Task Force (IETF) Internet-draft, June 1996.

[2] M. Khansari, A. Jalali, E. Dubois and P. Mermelstien, "Low bit-rate video transmission over fading channels for wireless microcellular systems," IEEE Trans. on CAS for Video Tech., pp. 1-11, Feb. 1996.

[3] E. Steinbach, N. Färber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environment," To appear in IEEE Trans. on CAS for Video Tech.

[4] M. Wada, "Selective Recovery of video packet loss using error concealment," IEEE Journal on Selected Areas of Communications, pp. 807-814, June 1989.

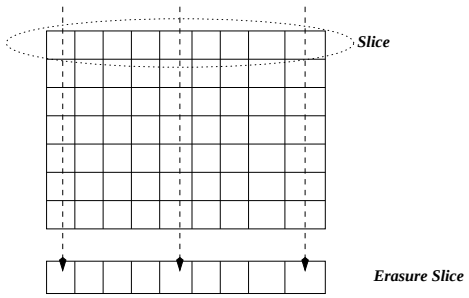

Figure 1: Picture format and Erasure slice

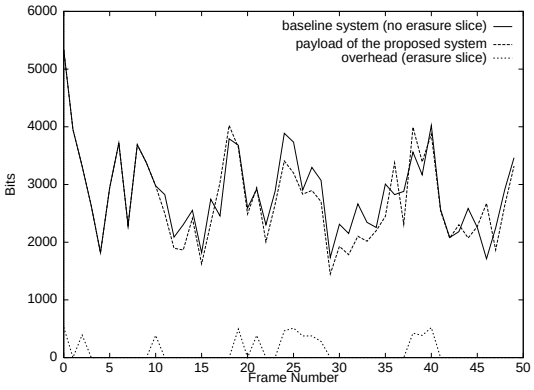

Figure 2: Bit sequences for the baseline and the proposed method

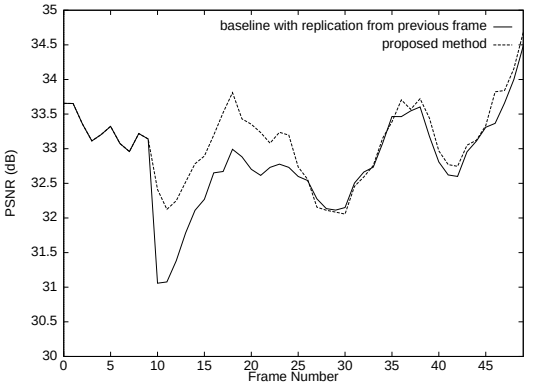

Figure 3: PSNR performance of the baseline and the proposed method

Figure 4: Reconstructed and residual error pictures for the baseline and proposed method (frame number 10)

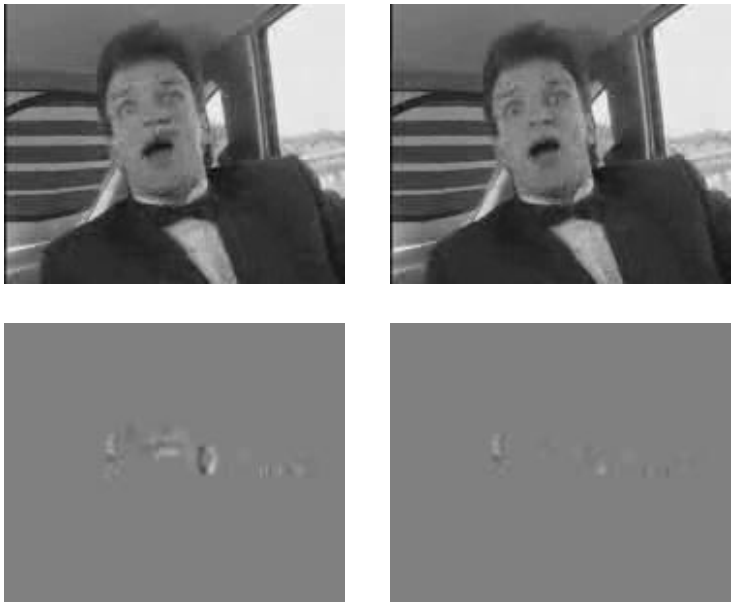

Figure 5: Reconstructed and residual error pictures for the baseline and proposed method (frame number 20)