ITU Telecommunication Standardization Sector
Study Group 15
Experts Group for Video Coding and Systems in
ATM and Other Network Environments

Document AVC-689
November 1994

SOURCE: BT Labs, UK
TITLE: Reduction of the bit-rate of compressed video while in its coded form
PURPOSE: For information

The following 4 pages were published as paper D17 at the Sixth International Workshop on Packet Video, Portland, Oregon, USA, 26-27 September 1994. The 5th page contains two additional SNR plots which were also presented at the workshop.

Paper D18, "VBR Transport of CBR encoded video over ATM networks" by Yong, Zhu and Eyuboglu of Motorola Information Systems Group contains a block diagram which is extremely similar to Figure 3 in the paper appended here.

Both papers could have relevance to the carriage of video over some types of Local Area Networks (LANs). However, BT is still of the opinion that H.320 over Isochronous Ethernet is the solution which ITU-T should recommend in H.32Z.

Reduction of the bit-rate of compressed video while in its coded form

D. G. Morrison, M. E. Nilsson and M. Ghanbari†

Centre for Human Communications, BT Labs, Martlesham Heath, Ipswich IP5 7RE, UK

†Dept. of Electronics Systems Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

Abstract. *A technique, operating on the coded data, for further reducing the bit rate of video already compressed with a predictive coding algorithm is introduced. Though illustrated with a motion compensated interframe hybrid DPCM/DCT algorithm, it is also generally applicable to other predictive coding schemes, the key elements being the discarding of some coded data, the calculation of the distortion introduced and the later inclusion of correction signals as a normal part of the lower rate data. Low additional delay and no new constraints on the originating coder are noteworthy characteristics of the method.*

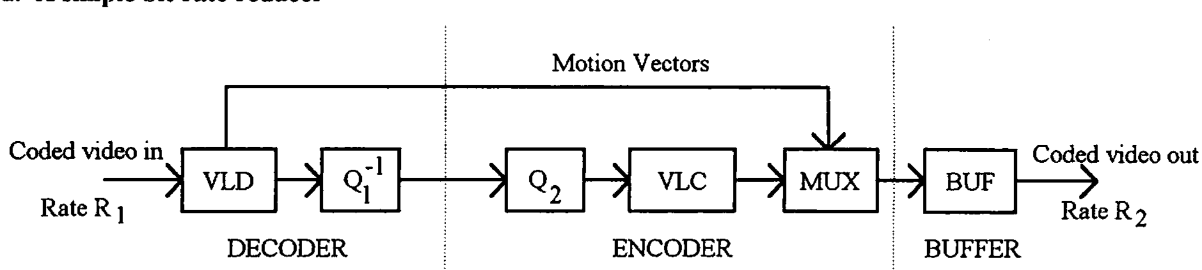
1. Introduction

The original motivation for our studies was to convey 64 kbit/s video between ITU-T Recommendation H.261 codecs over a 32 kbit/s link without modifying the codecs themselves in any way. The H.261 Recommendation has facilities, such as the Macroblock Address Stuffing codeword and empty Forward Error Correction blocks, which could be used to pad out the received 32 kbit/s data stream to generate a 64 kbit/s one which fully conforms to H.261 and is decodable by compliant decoders. Clearly, the difficult part would be the reduction from 64 kbit/s to 32 kbit/s without modifying, and preferably without even having access to, the originating encoder.

Though modifications of the video multiplex and use of different entropy coding techniques on the basic constituent H.261 data elements might achieve some modest lossless reduction of the bit rate, these approaches alone will not yield the required 2:1 change. Thus, some form of lossy compression or transcoding is needed.

The sources of the additional delay introduced by a transcoder between one compression algorithm and another had already been studied by Morrison. Pure processing delay arising from the physical implementation can be reduced to any target value, at least in theory, by faster hardware. When viewed separately, the delays through the receiving buffer of the decoding part and the rate smoothing buffer of the re-encoder appear to be inescapable. Generally, they are also dominant in magnitude compared to other irreducible latencies such as data reordering or block based operations. However, by considering the overall configuration of a transcoder, its total buffering, and hence delay, can be made substantially less than the sum of the apparent separate constituents [1]. Though such a transcoder is perfectably able to decode and then re-encode using the same algorithm but at a different rate, its major capability is the conversion from one algorithm to another. Since we did not need that feature, we wondered if another approach might be more suitable. Nevertheless, low additional delay remained a requirement.

2. A simple bit rate reducer



VLD:	Variable Length Decoder	Q_1^{-1}:	Inverse Quantiser
VLC:	Variable Length Coder	Q_2:	Quantiser
MUX:	Video Multiplexer	BUF:	Buffer

Figure 1. Basic strategy for bit rate reduction.

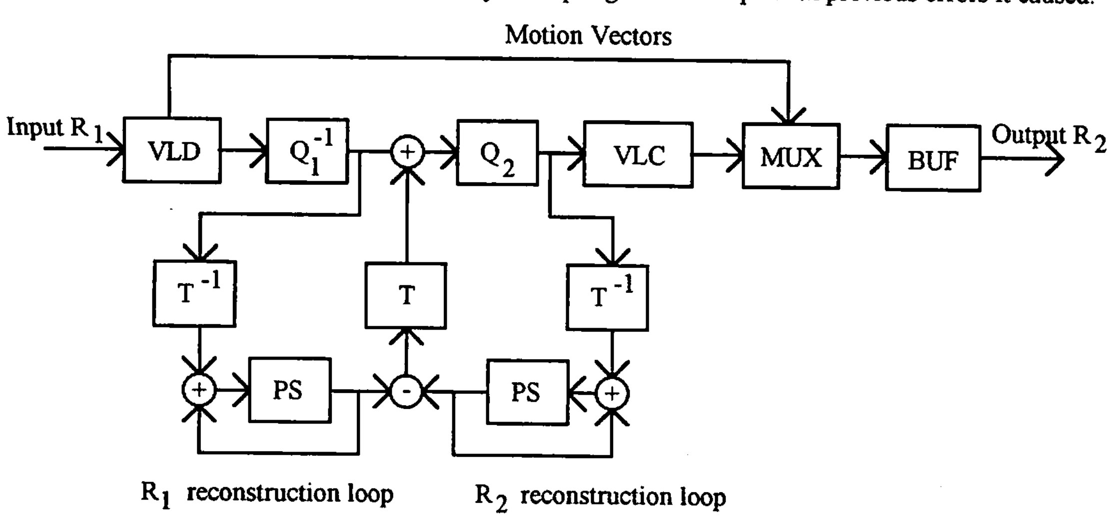
Dr. Ghanbari contributed to this work while on a Short Term Fellowship at BT Laboratories.

The fundamental strategy is to requantise the coded data with a larger step size. As illustrated in Figure 1 for a motion compensated hybrid DPCM/DCT scheme, the bit stream is decoded to motion vectors and transform coefficients. The latter are requantised and combined with the unmodified vectors to produce a new bit stream at a lower rate. Note that the decoding is performed on the coded data directly from the transmission system without buffering and that there is very little delay up to the output of the multiplexer because the processes have low latency. The only buffer is between the multiplexer and the second transmission system and serves the same two functions as in a normal encoder; namely smoothing the coded data and providing additional control of the quantiser Q_2. However, the smoothing required is only to cover any localised variations in output rate due to the discrete nature of Q_2, so this buffer and its delay can be an order of magnitude or more smaller than those in conventional encoders.

Although requantisation is satisfactory for PCM or intraframe DCT coding, there is a problem with predictive algorithms. Because requantisation is a lossy operation, the coding loops at the original encoder and the eventual decoder will contain different information. All predictive coding schemes have problems in such circumstances; the decoded images contain additional noise or artifacts which are likely to increase with time unless counter-measures are employed such as frequent reversion to non predictive (intra) coding, which opens the loop and discards the errors held in memory. As we wished to be able to accept incoming bit streams from unconstrained coders, we sought a solution to the drift problem which places no new requirements on them.

3. Bit rate reducer with drift correction

The solution, shown in Figure 2, is to apply a correction within the reducer itself, with the straight through path, at the top, unchanged. The concept is to requantise the transform coefficients as quickly as possible, accept that mistracking will be introduced between the original encoder and the eventual decoder, dispatch the requantised coefficients towards the eventual decoder, compute the error being introduced and attempt to correct it at the next opportunity. Of course, at that time a new error will be introduced by requantising the then current coefficients. Also, each correction can rarely be perfect given the discrete natures of the two quantising laws. Thus the reducer is continuously attempting to "catch up" with previous errors it caused.



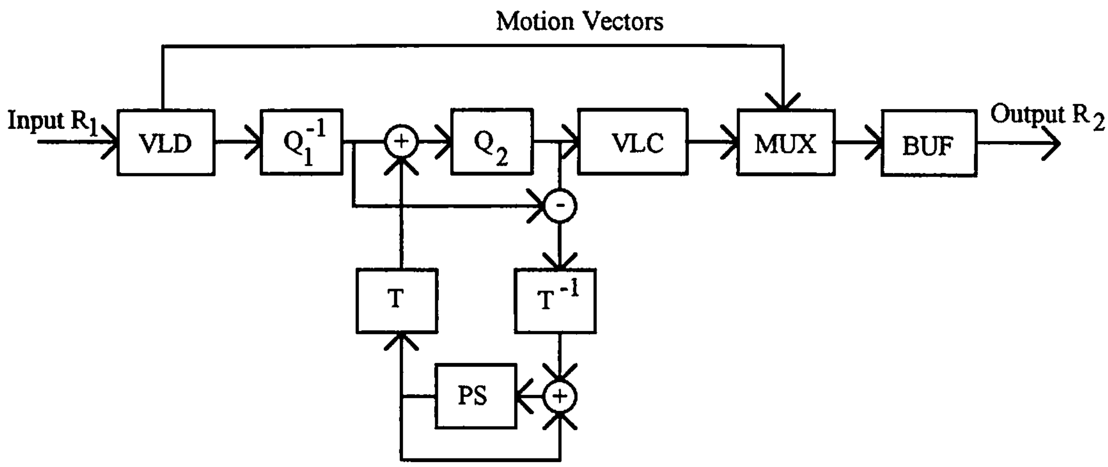
VLD:	Variable Length Decoder	Q_1^{-1}:	Inverse Quantiser
VLC:	Variable Length Coder	Q_2:	Quantiser
MUX:	Video Multiplexer	BUF:	Buffer
T:	DCT	T^{-1}:	IDCT
PS:	Prediction Store (Picture store, Frame store)		

Figure 2. Bit rate reduction with drift correction.

The error is obtained by fully decoding to the pel domain both the original coded data and the bit rate reduced version and forming the difference. This is then transformed and added back into the main path just before Q_2. Note that the reconstructed video signals used to form the error are taken from the outputs of the picture stores and are thus one picture late with respect to the main path.

4. Simplified arrangement

Figure 3 shows a rearranged configuration which is simpler to implement by virtue of eliminating one inverse DCT and one of the prediction stores. The straight through path is the same as before but the error is computed in the transform domain by differencing across the requantiser. Were it not for motion compensation, the correction signal could be formed in a loop operating in the transform domain. Instead, the error must be transformed to the pel domain and accumulated in that form. It is then subjected to a forward DCT to be suitable for use as the correction.



VLD:	Variable Length Decoder	Q_1^{-1}:	Inverse Quantiser
VLC:	Variable Length Coder	Q_2:	Quantiser
MUX:	Video Multiplexer	BUF:	Buffer
T:	DCT	T^{-1}:	Inverse DCT
PS:	Prediction Store		

Figure 3. Alternative configuration for bit rate reduction.

5. Performance

The scheme has been tested by simulation and an example set of luminance SNR plots for 74 pictures from the 'Miss America' sequence is given in Figure 4. The '64 kbit/s' and '32 kbit/s direct' curves are from a simulation of H.261. The '32 kbit/s from 64 kbit/s' curve is the result obtained by reducing the 64 kbit/s coded data. The first picture was coded entirely with intra macroblocks and is omitted from the plots. Intra refresh was disabled so that any long term degradation could develop uninterrupted.

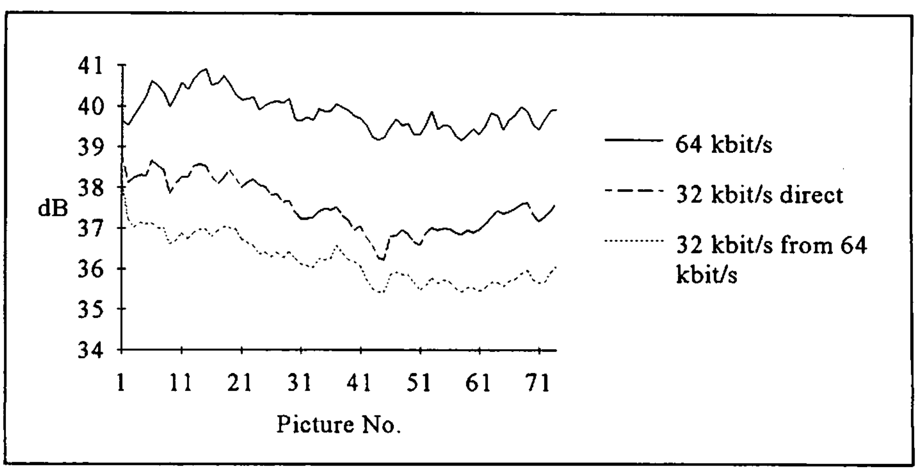


Figure 4. Simulation result with Miss America sequence.

As expected the results show that the SNR from the bit rate reduction method does drop for the first few pictures but then stabilises at 1 dB to 2 dB worse than the direct coding at the same final rate. Subjectively the distortions of the reduced version have the same character as those of the direct one but are slightly more noticeable. No new artifacts have been perceived.

6. Applications

The technique functions equally well in applications where the reduced rate is not constant. An example with special relevance to packet video, depicted in Figure 5, is a video connection between a terminal on the ISDN and another on a LAN. Compressed video from the ISDN terminal at a constant rate is transferred unchanged to the LAN by the gateway when the LAN traffic is sufficiently low. However, during periods of congestion on the LAN, the bit rate reducer in the gateway is able to reduce the video data rate required on the LAN. No control mechanism back to the original encoder is necessary. Thus the potential problems of transmission delay time to, and reaction time by, a distant encoder which might be anywhere in the world do not arise. Furthermore the reducer is able to make rate changes of almost arbitrary size and at arbitrary instants compared to, for example, the discrete 64 kbit/s or so steps at 20 millisecond boundaries possible in H.320/H.221.

Another example of fixed to variable bit rate conversion could arise in networks, such as for mobile applications, in which ARQ mechanisms dynamically reduce the effective throughput rate [2].

In both of the above application examples the reducer is normally inactive with $Q_2=Q_1$ and the picture quality is not impaired. The bit rate reduction is only invoked for comparatively short periods to mitigate against a transmission problem. Any temporary drop of picture quality is much more preferable than the usual effects of data loss on a predictive algorithm.

In many applications it may not be necessary to pad out the data back to the original bit rate.

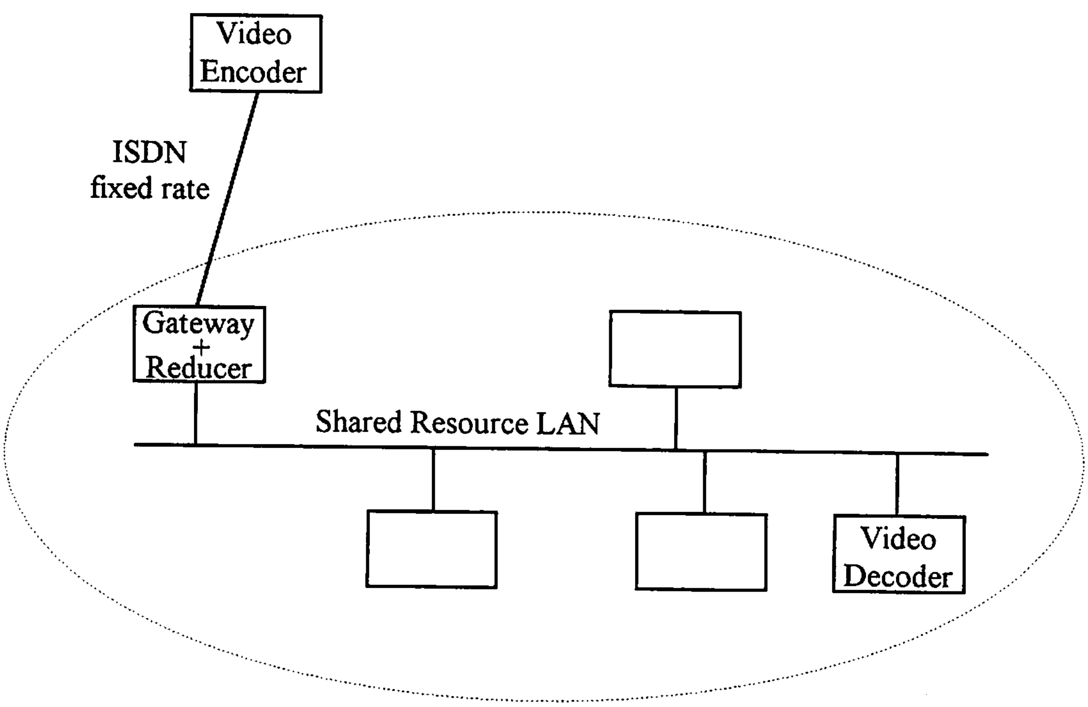


Figure 5. Bit rate reducer as part of an ISDN to LAN gateway.

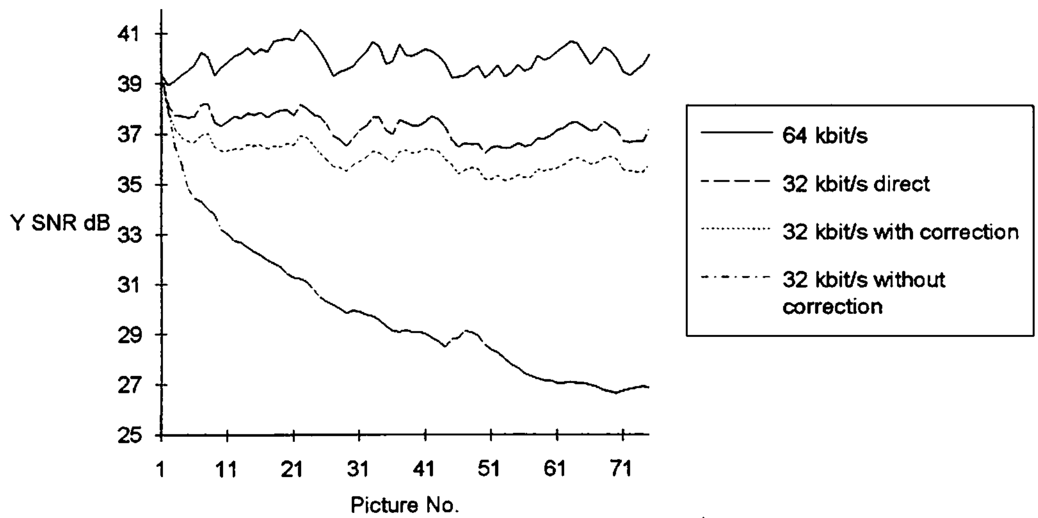
7. Conclusion

A technique has been described for further reducing the bit rate of compressed video data after the encoding has been performed. The method is applied to the data in its coded form and introduces minimal additional delay. It is especially suitable for providing temporary or variable reductions of coded bit rate. Most applications will be in gateways between networks.

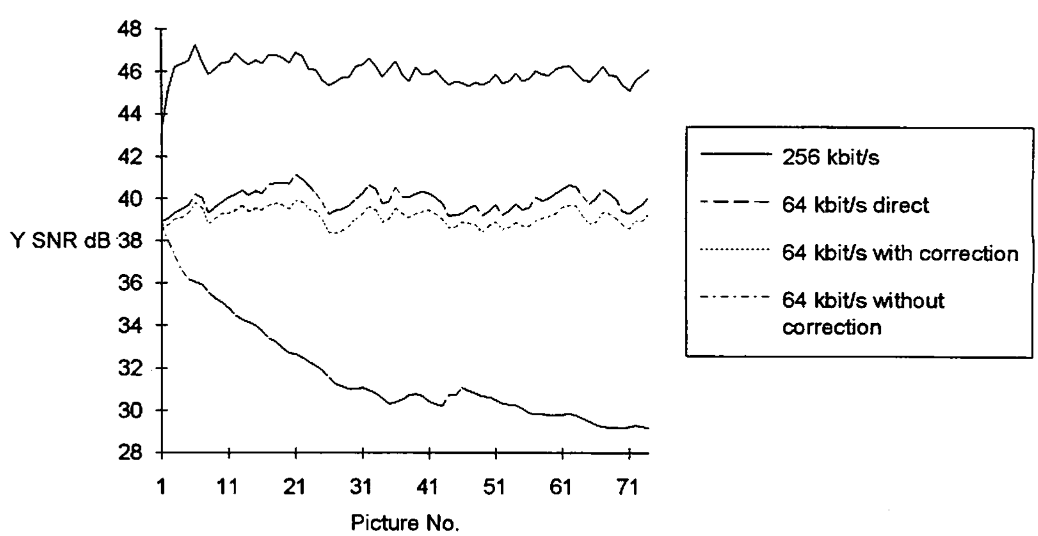
References

1. D. G. Morrison, 'Low delay video transcoders for multimedia interworking', First International Workshop on Mobile Multimedia Communications, Tokyo, 7-12 December 1993

2. N. E. MacDonald, 'Transmission of compressed video over radio links', BT Technology Journal, Vol. 11 No. 2 April 1993.



Simulation results for Claire sequence. H.261 64 kbit/s reduced to 32 kbit/s



Simulation results for Claire sequence. H.261 256 kbit/s reduced to 64 kbit/s