Multilayer Transcoding with format portability for multicasting of single-layered video

T. Shanableh*	M. Ghanbari
American University of Sharjah	University of Essex
Department of Computer Science	ESE Department
P.O. Box 26666, Sharjah, UAE	Essex, CO4 3SQ, UK
Email: tshanableh@aus.ac.ae	Email: ghan@essex.ac.uk
Fax +971 6 5585066	Fax +44 1206 872900

Abstract

This paper proposes a novel multilayer video transcoding approach for multicasting pre-encoded video to heterogeneous end-systems via diverse grouping of networks. Multilayer transcoding is first addressed by means of multi-quality or SNR scalability of the MPEG-2 standard. Frequency domain transcoding and drift-compensated transcoding are derived from the closed-loop and multi-loop SNR scalabilities respectively. The proposed transcoding architectures are verified in terms of eliminating picture drift whilst preserving compatibility with the MPEG-2 SNR decoder. Multilayer transcoding is then addressed by means of multi-resolution or spatial scalability of the MPEG-2 standard that supports different video formats. The transcoder retains the full resolution of the incoming video stream in its enhancement layer whilst generating a low spatio-temporal resolution base-layer compatible with the H.263 video format. Hence providing both multilayer transcoding and video format portability. The resultant video layers are shown to be free from drift with PSNR results comparable to those of the respective scalable encoders.

** Author for correspondence*

https://doi.org/10.1109/TMM.2004.840602

1. Introduction

a. Statement of the problem

A challenging application of video multicasting nowadays is the video distribution over heterogeneous networks that are connected together to form the branches of a multicasting tree. One example of a heterogeneous multicasting tree can be formed by transmitting video over the Internet Multicast Backbone or the MBone [1]. Similarly, in scenarios of distributing video for the purpose of distance learning or remote job training, users may be connected through various links and capacities. For instance, some users are connected through the corporate LAN others through xDSL, ISDN or maybe wireless/mobile links. Likewise, video distribution over a heterogeneous multicasting tree is found in applications of video multicasting in Cable Television Distribution Services (CATV). The ground station receives the pre-encoded video stream through a satellite link and distributes it through different capacity links onto heterogeneous end-systems [2].

For these applications of video multicasting, the question that this paper puts forward is as follows. If the distributed video is pre-encoded, how can the video source or distribution point adapt its transmission rate and/or approach in a manner commensurate with the multicasting tree networks or end-systems constraints?

To alleviate this problem, MaCanne proposed a Receiver-driven Layered Multicast (RLM) scheme [3]. The scheme proposes to encode video bitstreams in a multilayer fashion where it is then up to the receivers to select the appropriate number of video layers to subscribe to. This selection is based upon the underlying network-bandwidth constraints and therefore shifting the bitrate adaptability problem from the source to the individual receivers.

One of the primary advantages of the RLM is that it builds upon existing networks without imposing any extra functionality or speculations. However, the limitation of the RLM lies in the fact that the number of video layers or their bitrates might not suit the underlying multicasting tree. Therefore, pre-encoding video in a multilayer fashion can be counter-effective. Mainly, inefficiency manifests itself in terms of wasting bandwidth resources. Additionally, in a receiver-driven approach, all that a receiver can do is to

subscribe or unsubscribe to a video layer. This renders the adaptation granularity in the order of a video layer that might not result in a satisfactory bitrate adaptation.

Other solutions proposed to filter or transcode video bitstreams at various nodes of the multicasting tree such as routers, applications gateways or wireless base stations [4]. For instance, Hoffman proposed to implement a tag-based filtering at network routers. That is, to meet an instantaneous network congestion or limitation, a network node can start by dropping B-pictures of an MPEG-1 video. If the constraint is still not met, further P-pictures are dropped and so forth [5]. A similar filtering technique was also proposed by [6].

Despite its simplicity, one of the drawbacks of the tag-based filtering is again, the inaccurate or coarse bandwidth adaptation. To overcome this problem, Assuncao and Ghanbari proposed to insert a homogeneous video transcoder that implements bitstream re-quantization with drift-correction at those network nodes connected to constrained paths [7]. The idea of re-encoding or transcoding at network nodes was subsequently emphasized by [8] and [9].

Unlike the receiver-driven approach, video transcoding at various network nodes makes no assumptions regarding the multilayer coding of the source video and therefore providing a loose coupling between the network constraints and the video source. Nevertheless, such an approach has the following shortcomings. First, current networks routers do not support video transcoding or re-encoding. Second, for a heterogeneous multicasting tree comprising a diversity of bandwidth constraints, the number of video transcoders can grow significantly and hence; the need to transcode the video more than once alongside a transmission path. This multi generation transcoding results in a continuous drop in video quality [10].

b. The proposed multilayer transcoding solution

Based on the above discussion, the solution proposed in this paper brings together the advantages of both video transcoding and multilayer coding. The solution is concerned with fine bitrate adaptation without imposing any extra functionality on the underlying networks. Likewise, the solution does not

impose any speculations on the scalability parameters of the pre-encoded video. The solution put forward is a *single-to-multilayer video transcoder* that can be incorporated at the video source or the video distribution point rather than the network nodes. An illustration of the multilayer transcoder for video distribution through *IP multicast* is shown in Figure 1.

Hence, the number of video layers and their transmission rates can be adjusted dynamically to adapt to the structure or even better, the dynamic structure of the underlying multicasting tree. Moreover, by transcoding a coded video into multilayers, where each video layer is generated with respect to an underlying constraint, the proposed transcoder can substitute several single-layer transcoders.

Although multilayer video transcoding was originally proposed by the author in [11], nevertheless a number of interesting similar applications have been recently reported. Worth mentioning is a *single-to-FGS* transcoding technique proposed for elastic storage of compressed multimedia content. The MPEG-4 FGS scalability provides for DCT bit-plane coding of the enhancement layer that can be truncated according to the target bitrate eliminating the need for multiple retranscoding [12]. Likewise an inverse technique of *FGS-to-single* layer transcoding is reported to support non-FGS enabled end-systems. This is important since scalable receivers did not *yet* gain wide market deployment due to the increased computations and memory requirements [13].

In this paper, two categories of multilayer transcoders are derived. In the first category, the derivation is based upon the closed-loop and the multi-loop SNR scalability [14] and [15]. The objective is to generate a multi-quality and multi-bitrate video stream. The other category is based upon the MPEG-2 spatial scalability with the objective of generating different spatio-temporal resolutions whilst supporting video format portability.

2. MPEG-2 *SNR* scalability

The video scalability defined in the MPEG-2 standard can be divided into three categories. Namely, multi-quality scalability including *data partitioning* and *SNR* scalability, multi-resolution scalability including *spatial and temporal* resolution scalability and finally, a combination of these different scalabilities is also supported and is referred to as *hybrid* scalability.

The SNR scalability was originally proposed by Ghanbari in [14] to increase the robustness of video streaming against cell loss in ATM networks. More recently, the SNR scalability found its way to other important applications such as providing multi video qualities at different bitrates.

The MPEG-2 standard defines the SNR decoder as illustrated in Figure 2. The dequantized DCT coefficients of all the incoming SNR layers are added to those of the base layer, inverse transformed and fed into the motion compensation (*MC*) loop of the decoder.

To comply with this decoder and depending on the target application of concern i.e., error resilience or provision of a multi-quality bitstream, two SNR encoding architectures are employed. In one, the enhancement layers are formed by requantizing, using a finer quantizer step size, the quantization error of the previous layer. Subsequently, the sum of dequantized coefficients of all the enhancement layers is added to the *MC*-loop of the base layer. This encoder, referred to as a *closed-loop* encoder, is illustrated in Figure 3.

While in the other SNR encoding architecture, the error picture formed by subtracting the locally reconstructed from the original/input picture of the previous layer is fed to another MPEG-2 encoder. This *multi-loop* encoder reuses the motion vectors of the base layer in all of the *MC* loops of the subsequent enhancement layers as shown in Figure 4. The architecture verification of the two encoders is elaborated upon in Appendix A.

On the other hand, note that open-loop SNR encoders are not compatible with MPEG-2. In such architecture, the enhancement layers are coded without motion compensation and their dequantized coefficients are not fed back to the base layer [15]. This is similar to base-line FGS coding in MPEG-4. It follows that video layers are decoded separately and then added to produce the final picture which deviates form the standardized MPEG-2 decoder of Figure 2.

3. Single-to-multilayer SNR transcoding

3.1 Drift Compensated Multilayer (DCM) transcoding

In video transcoding it is desirable that each video layer can be decoded correctly without any picture drift, hence in DCM transcoding each video layer should have its own drift correction MC-loop. This transcoding architecture is

shown in Figure 5.

Drift correction is provided as follows. The input coefficients to the enhancement layer are subtracted from those obtained via the inverse quantizer. This difference signal, which represents the information lost in the transcoding process, after conversion from the frequency domain to the pixel domain by an inverse discrete cosine transform (DCT), is then accumulated in a MC-loop which receives motion vectors extracted from the incoming signal. This accumulated drift is then converted to the frequency domain and added as a correction to the next frame. Note that the frame store is shown as multiple stores, which are needed in systems such as MPEG which use an irregular inter-frame prediction sequence. The arrangements for switching between the stores are as in a conventional MPEG coder. For further information on drift correction in video transcoding please refer to the co-author's work in [16].

Appendix B verifies the drift free property of the proposed transcoder and elaborates upon its full compatibility with the MPEG-2 SNR decoder.

Note that by employing a switch at the *MC* loops of the multi-loop transcoder, the feedback loops can be disabled. In this case, the DCM architecture of Figure 5 is simplified as shown in Figure 6. This illustrates a three layer SNR scalable transcoder where drift correction is removed completely. This would only be suitable for applications where regular intra refresh pictures are transmitted (as in many MPEG applications, where it would rely on the periodic I- pictures of the incoming video stream, otherwise the resultant drift may render the picture quality unacceptable). This architecture is referred to as Non-Drift Compensated Multilayer (NDCM) transcoder.

Clearly, such a switch can operate in according to the *GoP* structure of the incoming video stream. If periodic intraframe coding is employed then picture drift becomes less of an issue regardless of whether or not drift compensation is employed.

The difference between the two transcoders resembles the difference between multi-loop and closed-loop encoding in MPEG-2 in a number of features. For instance the use of MC-loops in the enhancement layers of the DCM transcoder without the feedback into the base layer renders it similar to

the multi-loop encoder. On the other hand, similarity between the_NDCM transcoder and the closed-loop encoder is due to the absence of the MC-loops. Note that in a brute-force method of transcoding involving a cascaded decoder and a closed-loop encoder (shown in Figure 7), the feedback of the dequantized coefficients of the enhancement layers into the base reduces the difference between the incoming picture and the locally decoded one. That is, the difference between the two is now due to the quantization error of the top enhancement rather than the base layer. As such, the motion-compensated pictures of the decoder and encoder parts of the cascaded transcoder become nearly identical. Hence the two MC-loops can approximately cancel each other out, resulting in the simplified transcoding architecture of Figure 6. This remark further supports the similarity between the NDCM transcoder and the closed-loop encoder.

In Figure 7, it is shown that the subtraction of the two motion compensated pictures at point '*S'* can be approximated to nil resulting in the aforementioned simplified transcoding architecture. The cascaded architecture of the base layer is adopted from [17]. Again, formalization and notations are given in the appendices.

3.2 Experimental results:

One of the main features of video transcoding is to meet or respond to bandwidth constraints. It follows that video transcoders should incorporate adequate bitrate control algorithms. In the following experiments, two different algorithms are adapted from the authors' previous work. The first algorithm distributes bits among macroblocks according to the instantaneous buffer fullness with the goal of achieving similar picture quality to that produced by direct encoding (refer to [2] for a detailed derivation). Whereas the second algorithm considers an optimal approach to bit allocation with the goal of minimizing the average overall distortion of transcoded pictures. Again, the particulars of the bitrate control algorithms are outside the scope of this paper. Interested readers shall refer to [16] for more information. Both algorithms are adapted in the base and enhancement layers of the proposed transcoders respectively. The SNR encoders are extended from the publicly available MPEG Software Simulation Group (MSSG) codec that implements TM5

bitrate control [24].

In Figure 8 the incoming video is coded with a GoP structure of (N = 12, M = 3) at 4Mbit/s. The video is transcoded into 4 layers at 1 Mbit/s per layer. The figure shows that the NDCM transcoding results coincide with those of the closed-loop SNR encoder.

Recall that in the closed-loop encoder, the stand-alone decoding of the base layer causes a mismatch between the encoder's and the decoder's MC-loops as detailed in Appendix A. This should be distinguished from the mismatch caused by requantizing the DCT coefficients in the NDCM transcoder. The extent of the MC-loop mismatch varies depending upon the underlying video sequence and coding/transcoding parameters. Hence if for a given coded picture the mismatch caused by the NDCM transcoder is finer than that of the closed-loop SNR encoder, the transcoded pictures may have a higher quality. This is noticed in part a of Figure 8.

Moreover, the figure shows that single-layer encoding outperforms SNR multilayer coding. This statement holds for the MPEG-2, MPEG-4 and the H.263 + SNR scalabilities as reported in [18], [19] and [20] respectively.

On the other hand, although SNR scalability is inferior to single layer coding, one can argue that such a comparison is unfair. Other factors should be taken into account, for instance, when multilayer coding is employed instead of simulcast, the bit-rate saving can be significant. In Figure 8 if the four video layers are transmitted independently at 1,2,3 and 4Mbit/s then the total bit-rate mounts up to 10Mbit/s, which is a 150% increase in bit-rate over the multilayering case.

Additionally note that multilayer transcoding comes with the advantage of generating the appropriate number of video layers and bit-rate in a manner commensurate with the underlying network-bandwidth and end-system constraints. Whereas in the case of pre-encoded multilayer video, coarse assumptions about the scalability parameters are likely to take place.

In Figure 9 it is shown that the additional complexity of the feedback loops of the DCM transcoder comes with the advantage of drift-free transcoding of the base layer. On the other hand, the simplicity of the NDCM transcoder is supported by the superior picture quality of the enhancement layer. The difference between the picture qualities follows from the difference between

the closed-loop and the multi-loop SNR scalabilities in MPEG-2 as illustrated in [21] and [22]. In the former it is known that the efficiency of the enhancement layers comes at the expense of introducing picture drift artifacts into the base layer. Part a of the figure shows that if periodic I-frames are employed then NDCM transcoding can still produce acceptable picture quality at the base layer. On the other hand, the absence of periodic I-frames can cause severe picture drift as shown in Part b of the figure. The figure also shows that stand-alone decoding of the base layer in the closed-loop SNR encoding causes a similar picture drift.

To verify further the above discussion, table 1 summaries the multilayer encoding and transcoding results for three additional test sequences. As shown in the table, the quality differences among various encoding and transcoding architectures are marginal. Nevertheless the following facts are evident. First, the base layer quality of closed loop encoding is lower than its multi-loop counterpart. This statement also holds for the difference between NDCM and DCM transcoding. Second, the enhancement layer quality of closed loop encoding is higher than its multi-loop counterpart. Again, this statement also holds for the difference between NDCM and DCM transcoding.

.

A trade-off between drift free decoding of base layers and simplified transcoding architecture is achieved by disabling the drift-compensation loops for the transcoding of the B-pictures The reason behind this is that B-pictures themselves are not used as a source of prediction and therefore any deficiency in transcoding them without drift correction persists for their duration only and do not propagate into future pictures. Additionally, considering the typical *GoP* structures of MPEG-2 video of N = 12, M = 3 which contains 8 B-pictures, disabling the feedback loops for B-pictures results in significant savings in feedback loop complexity.

4. Spatio-temporal scalability transcoding

The bitrate reduction of the introduced SNR transcoders is achieved by requantizing the incoming DCT coefficients. It follows that if a harsh bitrate reduction is required for the base layer then the spatio-temporal resolution of the incoming stream has to be reduced. Moreover, end users or systems with

low bitrate requirements will probably have other types of video decoders such as the H.26x family decoders. In such a scenario, a multilayer transcoder is required to carry out two different tasks for the base layer. In one, the base layer is expected to have a low bitrate with a low spatio-temporal resolution to accommodate lower capacity networks/connections or end users capacity/processing power. While in the other, the base layer is expected to comply with a different encoding format. Fortunately the loose-coupling property between video layers in the MPEG-2 spatial scalability provides for portability between different encoding formats. In the MPEG-2 spatial scalability, the enhancement layer can be predicted from the locally decoded pictures of the base layer. Likewise, the enhancement layer can be decoded by reconstructing it from the decoded pictures of the base layer. This implies an interesting feature, the base layer need not conform to the encoding format of the enhancement layer. This is because the interface between the two layers is restricted to the pixel-domain decoded pictures of the base layer. For instance, Figure 10 shows an MPEG-2 complaint spatial decoder with different encoding formats for the base and the enhancement layers. Once a base layer picture is fully decoded, it is spatially interpolated and used in the reconstruction of the enhancement layer. Note that three prediction modes are available for the enhancement layer; independent temporal prediction, spatial prediction from the base layer or a weighted combination of temporal and spatial prediction.

4.1 Derivation

Since transcoding of MPEG-1,2 video into the H.263 format is already established as proposed by the authors in [23], the output of the base layer of the spatio-temporal scalability (*STS* for short) transcoder can now conform to the H.263 format. Furthermore, this base layer can be simplified by skipping the B-pictures of the incoming video stream resulting in low-complexity and low-delay transcoding with lower spatio-temporal resolution. The enhancement layer on the other hand has to retain the encoding format and the full spatio-temporal resolution of the incoming stream whilst allowing some flexibility in reducing the total bitrate by means of quality reduction or in other words by means of re-quantization.

To meet these tasks, the transcoding architecture of Figure 11 is proposed. In the figure, the incoming video stream is decoded up to the DCT coefficients and sent to both the enhancement and the base layers. Again, temporal scalability is achieved by skipping B-pictures in the base layer. The remaining I and P-pictures are transcoded into the H.263 format with quarter of the incoming spatial resolution. The locally decoded pictures are then interpolated and subtracted from the incoming ones to form a potential source of prediction for the corresponding pictures of the upper layer. Lastly, the resultant prediction error is re-quantized to control the overall bitrate. Consequently, a drift correction *MC* loop is employed to prevent the final decoded pictures from any drift.

Since the spatial prediction is temporally independent, a spatially predicted macroblock will cause the respective location in the frame buffer of the *MC* loop to be flushed or replaced just as the case with intraframe coding. Obviously since this spatially predicted macroblock is re-quantized, the resultant quantization error is motion compensated and added to the next picture that uses it as a source of prediction.

On the other hand, as verified by the experimental results of the next section, by increasing the percentage of the spatial prediction (i.e. prediction from the base layer) the temporal dependencies between the enhancement layer pictures are reduced. Hence the need for the drift-correction *MC*-loop becomes less demanding. This results in a simplified transcoding architecture as illustrated in Figure 12.

Various techniques may be employed for increasing the percentage of spatial prediction in the enhancement layer. For instance, the variance of a macroblock spatial prediction can be deliberately decreased by a constant percentage prior to comparing it against that of the temporal prediction. Hence biasing the macroblock mode decision towards spatial rather than temporal prediction. A similar technique of interfering with the macroblock mode decisions was successfully applied by the authors for error resilient coding as introduced in [25].

4.2 Experimental results

In this section, common to all of the following experiments, unless otherwise stated, the test sequences are MPEG-2 coded at 1.5Mbit/s (SIF) with a *GoP*

structure of N=12 and M=3. The coded sequences are then transcoded into two layers. The base layer is H.263-coded at 140kbit/s[1]. The bitrate of the enhancement layer is 850kbit/s obtained by the bitrate control algorithm outlined in section 3.2 [2]. Throughout the experiments, the spatial scalability results are compared against transcoding of the same sequence using a single-layer transcoder with drift-correction similar to that of the enhancement layer. The bitrate of the single layer transcoder is made equal to the total bitrate of the spatial transcoder i.e. just under 1Mbit/s.

To validate the transcoder of Figure 11, the picture quality of the enhancement layer is compared against the single-layer MPEG-2 transcoder as shown in Figure 13. The slight degradation in quality is justified by the excessive overhead of the spatial scalability syntax including the sequence header extension, picture header extension and the macroblock encoding modes extension. Nevertheless, since I-pictures are now allowed to choose between intraframe and spatial prediction, the figure shows that their quality peaks over those of the single-layer transcoder. The figure also plots the quality of the QCIF pictures of the H.263 base layer (compared against down-sampled uncompressed pictures) with a frame rate of 25/(M=3).

Since the spatial prediction from the base layer is temporally independent , it follows that with the simple re-quantization of the enhancement layer of Figure 12 one would expect less drift propagation compared to single-layer non-drift compensated transcoding. Clearly, the picture-drift reduction is proportional to the percentage of spatially predicted macroblocks in interframe coded pictures (P-pictures). Hence, In the transcoding architecture of Figure 12, the higher the percentage of spatially predicted macroblocks the lower is the resultant picture-drift. To illustrate this, the percentage of spatially predicted macroblocks in both transcoding architectures of Figure 11 and 12 are investigated. In the former architecture, since the drift of transcoded pictures is added to the next picture that uses the current one as a source of prediction, the sum of absolute values of the resultant DCT coefficients is expected to rise. Whereas in the latter transcoder, since no drift correction is

[1] Telnor's implementation of H.263 and TM5 bit rate control is adopted in this work.

employed it follows that the sum of the absolute values of the transcoded coefficients is expected to decrease in the sense that coarser quantization might force more coefficients to zero. In this case the incoming interframe macroblocks have a higher probability of preserving their temporal prediction rather than substituting it with a spatial prediction. In contrast, with the drift-correction loop of Figure 11, the transcoder is expected to decide on higher percentage of spatially predicted macroblocks. This observation is verified in Figure 14. The figure plots the percentage of spatially predicted MBs in the two architectures for I and P pictures. Note that since B-pictures are skipped in the base-layer to provide temporal scalability, it follows that the enhancement layer cannot employ spatial prediction for such a picture type. Hence B-pictures are not considered in the figure.

Since I-pictures are not compensated for drift, the percentage of spatially predicted macroblocks in both architectures is equal. In both cases it is understood that spatial prediction is preferred over intraframe coding since MPEG-2 allows spatially predicted macroblocks in I-pictures to be skipped. As for P-pictures and in comparison with the drift compensated architecture, it is evident that the percentage of spatially predicted macroblocks without drift compensation is less than 50%. This low percentage is far beyond what is needed for suppressing picture-drift. However, as described in the previous section, this low percentage can be deliberately increased by interfering with the macroblock mode decision [25]. To achieve this, the variance of the spatial prediction is decreased by a constant percentage prior to comparing it against that of the temporal prediction.

In Figure 15 a *GoP* structure of N= ∞ and M=3 is employed. The three test sequences are transcoded without drift-compensation. The figure shows that by increasing the percentage of spatially predicted macroblocks, the overall quality becomes superior to that of the single layer transcoder. Note that by forcing all the interframe macroblocks to be spatially predicted, the overall quality slightly degrades. Unlike I-pictures, the MPEG-2 standard does not allow skipping spatially predicted macroblocks in P-pictures. Therefore, what used to betemporally-predicted and skipped macroblock is now transcoded with spatial prediction. Likewise, some interframe macroblocks might be

coded efficiently without any prediction error and so forth. As a result of forcing such macroblocks to be spatially predicted, a higher number of bits is needed leading to a coarser quantization and a lower decoding quality. Nevertheless, Figure 15 shows that the overall quality is still superior to that of the single layer transcoder. To sum up, transcoding by means of simple re-quantization without drift-correction of the enhancement layer can become feasible provided that the percentage of spatially predicted macroblocks is increased.

5. Conclusions

We have devised a number of novel single-to-multilayer transcoders that have applications to multicasting pre-encoded video over heterogeneous networks and end-systems. The solution has a number of advantages namely; storage space is saved in comparison to employing replicated video streams, bandwidth resources are saved in comparison to employing multiple single layer transcoders and finally, transmission flexibility is gained in comparison to employing pre-encoded multilayer video . This flexibility is in terms of the number of generated video layers and their bit-rates.

Three categories of video transcoders were proposed. In the first, a multilayer transcoder that employs simple requantization without drift correction is derived from the closed-loop SNR scalability. This simple transcoder benefits from the periodic I-pictures of the incoming video streams for minimizing the effect of picture drift.

In the second, a multilayer transcoder is derived from the multi-loop SNR encoder hence, all video layers are free from picture drift. This drift free transcoding comes at the expense of slight quality degradation in the enhancement layers when compared to the first transcoder.

Furthermore, it was shown that by disabling the drift-compensation loops for B-pictures, the transcoding architecture is simplified at the expense of a slight but local quality degradation in the base layer.

Lastly, it was shown that the spatio-temporal scalability (*STS*) transcoder generates a low bitrate, low resolution base layer that supports format portability. The full spatio-temporal resolution of the incoming video is preserved in the enactment layer of the *STS* transcoder whilst controlling its bitrate by means of requantization., It was also shown that by increasing the

percentage of spatial over temporal predictions, the temporal propagation of drift in the enhancement layer is suppressed. Therefore, the enhancement layer can be formed by a simple re-quantizer without drift tracking and correction. The base layer on the other hand benefited from the loose coupling between video layers in the spatial scalability and hence was transcoded into a different video format.Acknowledgements:
The authors wish to acknowledge the financial support of the Engineering and Physical Science Research Council (EPSRC) of the U.K. The works was also partly funded by BTexact Technologies, UK.

References

[1] S. E. Deering, “Internet multicast routing: state of the art and open research issues,” *Multimedia Integrated Conferencing for Europe (MICE),* Seminar at the Swedish Institute of Computer Science, October 1993

[2] P. A. A. Assuncao and M. Ghanbari, “Buffer Analysis and Control in CBR Transcoding,” *IEEE Trans. Circuits and Systems for Video Technology*. 10(1), pp. 83-92, February 2000

[3] S. McCanne, V. Jacobson and M. Vetterli, “Receiver-driven layered multicast,” *Proc. of ACM SIGCOMM’96*, Stanford CA, August 1996

[4] E. Amir and S. McCanne, “An application level video gateway,” *Proc. of ACM Multimedia ‘95,* San Francisco CA, November 1995

[5] D. Hoffman, M. Speer and G. Fernando, “Network support for dynamically scaled multimedia data streams,” *Proc. of 4th Inter. Workshop on Network & Operating Systems Support For Digital Audio & Video*, Lancaster University, UK, 1993

[6] M. Hemy, U. Hengartner, P. Steenkiste and T. Gross, “MPEG System Streams in Best-Effort Networks,” *Proc. of Packet Video ’99*, New York, April 1999.

[7] P. Assuncao and M. Ghanbari “Multi-casting of MPEG-2 video with multiple bandwidth constaints,” *Proc. of the seventh international workshop on Packet Video*, pp. 235-283, March 1996.

[8] R. Ahlswedw, N. Cai, S.-Y. R. Li and R. W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, 46(4), pp.1204-1216, July 2000

[9] S. D. Servetto and M. Vetterli, “Video multicast over fair queueing networks,” *Proc. of ICIP2000, Vancouver*, September 2000.

[10] O. Werner, “Requantization for transcoding of MPEG-2 video intraframes,” *IEEE Trans. Image Processing*, 8(2), pp. 179-191, February 1999

[11] T. Shanableh, “Heterogeneous video transcoding for matching network-bandwidth and end-system constraints,” *PhD thesis, ESE department, University of Essex*, UK, October 2001.

[12] E. Barrau, "MPEG video transcoding to a fine-granular scalable format," *Proc. IEEE Int'l Conf. Image Processing*, Rochester, NY, September 2002.

[13] Y.C. Lin, C.N. Wang, T. Chiang, A. Vetro and H. Sun, "Efficient FGS-to-Single layer transcoding," *Proc. IEEE Int'l Conf. Consumer Electronics*, Los Angeles, CA, June 2002.

[14] M. Ghanbari, “Two-Layer coding of Video signals for VBR networks,” *IEEE journal on Selected Areas in Communications,* 7(5), pp. 771-781, June 1989

[15] M. Ghanbari, “An adapted H.261 two-layer video codec for ATM networks,” *IEEE Trans. Communications*, 40(9), pp. 1481-1490, September 1992

[16] P. A. A. Assuncao and M. Ghanabari, “A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams,” *IEEE Trans. Circuits and Systems for Video Technology*, 8(8), pp. 953-967, December 1998

[17] P. A. A. Assuncao and M. Ghanbari, “Transcoding of single-layer MPEG video into lower rates,” *IEE Proceeding on Vision, Image and Signal Processing*, 144(6), pp. 377-383, December 1997

[18] J. F. Arnold, M. R. Frater and Y. Wang, “Efficient drift-free signal-to-noise ration scalability,” *IEEE Trans. Circuits and Systems for Video Technology*, 10(1), pp. 70-82, February 2000

[19] W. Li, “Overview of fine granularity scalability in MPEG-4 video standard,” *IEEE Trans. Circuits and Systems for Video Technology*, 11(3), pp. 301-316 March 2001.

[20] M. Walker and M. Nilsson, “A study of the efficiency of layered video coding using H.263,” *Proc. of International workshop on Packet Video, PV ’99*, New York, April 1999.

[21] [Mathew and Arnol, 1997] R. Mathew and J. F. Arnold, “Layered coding using Bitstream Decomposition with Drift correction,” *IEEE Trans. Circuits and Systems for Video Technology,* 7(6), December 1997 [22] R. Mathew and J. F. Arnold, “Layered coding using Bitstream Decomposition with Drift correction*,” IEEE Trans. Circuits and Systems for Video Technology*, 7(6), December 1997

[23] T. Shanableh and M. Ghanabari, “Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats,” *IEEE Tran. Multimedia*, vol. 2, no. 2, pp. 101-110 June 2000.

[24] Test Model 5, ISO/IEC JTC1/SC29/WG11/ N0400, MPEG93/457, April 1993

[25] T. Shanableh and M. Ghanbari, "Backward tracking of B-pictures bidirectional motion for interframe concealment of anchor pictures," *Proc. of ICIP2000*, Vancouver, September 2000

Figures and tables of Manuscript: Multilayer Transcoding with format portability for multicasting of single-layered video.

Two layer encoding and transcoding								
	Base layer PSNR [dB]				Enhancement layer PSNR [dB]			
Seq-uence	**CL-ENC-B**	**ML-ENC-B**	**NDCM-B**	**DCM-B**	**CL-ENC-E**	**ML-ENC-E**	**NDCM-E**	**DCM-E**
Sales Man	34.2	35.2	33.4	34.5	38.5	37.7	36.8	36.6
Coast Guard	30.3	31.2	29.7	30.7	34.1	33.6	33.0	32.8
Table Tennis	31.5	32.4	31.5	32.3	36.0	35.3	34.4	34.3
Acro-nyms	**CL-ENC-B:** Closed loop multilayer encoding – base layer **ML-ENC-B:** Multi-loop multilayer encoding – base layer **NDCM-B:** Non-drift compensated multilayer transcoding – base layer **DCM-B:** Drift-compensated multilayer transcoding – base layer **CL-ENC-E:** Closed loop multilayer encoding – enhancement layer **ML-ENC-B:** Multi-loop multilayer encoding – enhancement layer **NDCM-B:** Non-drift compensated multilayer transcoding – enhancement layer **DCM-B:** Drift-compensated multilayer transcoding – enhancement layer							

Table 1. Two layer encoding and transcoding for three test sequences. Video layers are coded at 0.75Mbit/s, 25Hz with a GoP of N=12, M=3.

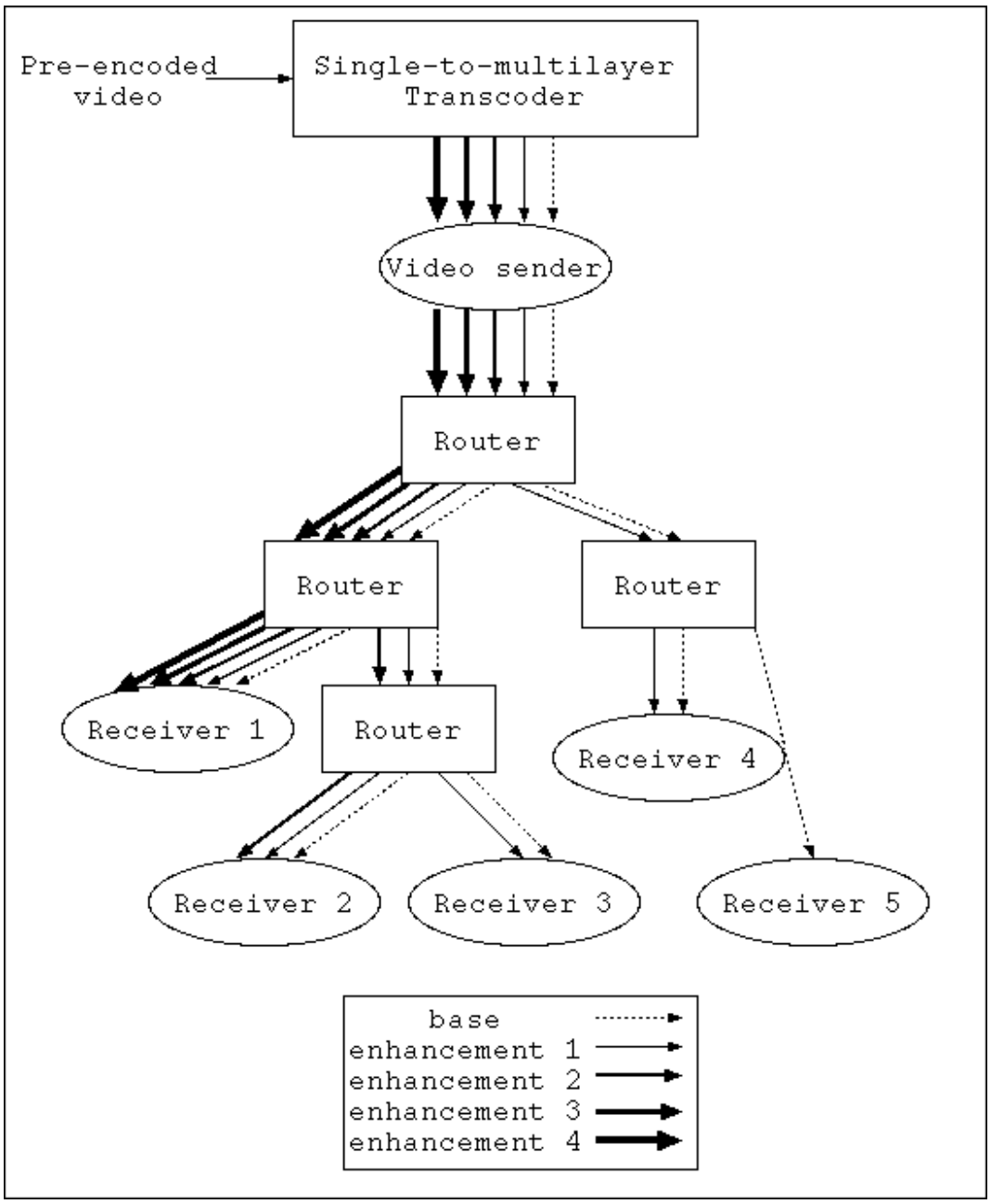


FIGURE 1. Multilayer video transcoding for multicasting of single-layered video.

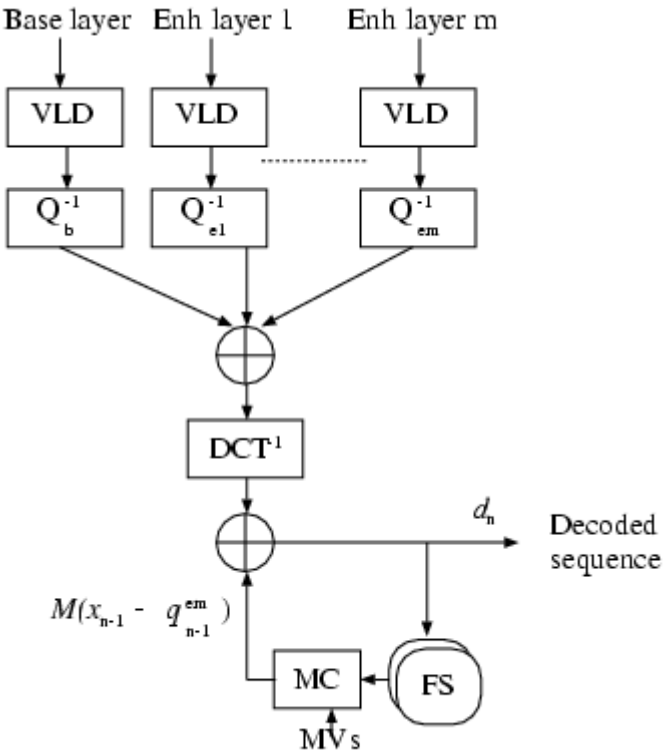


Figure 2. Block diagram of MPEG-2 SNR decoder.

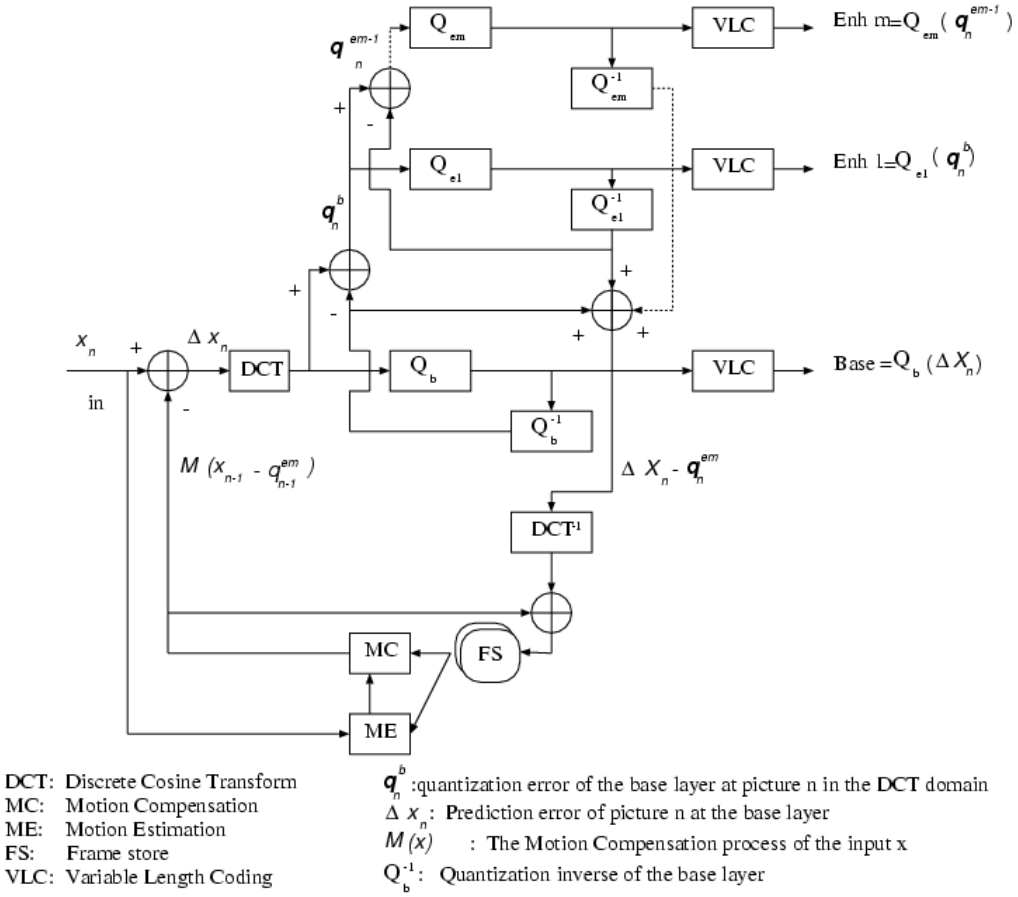


Figure 3. Block diagram of *closed-loop* SNR encoder

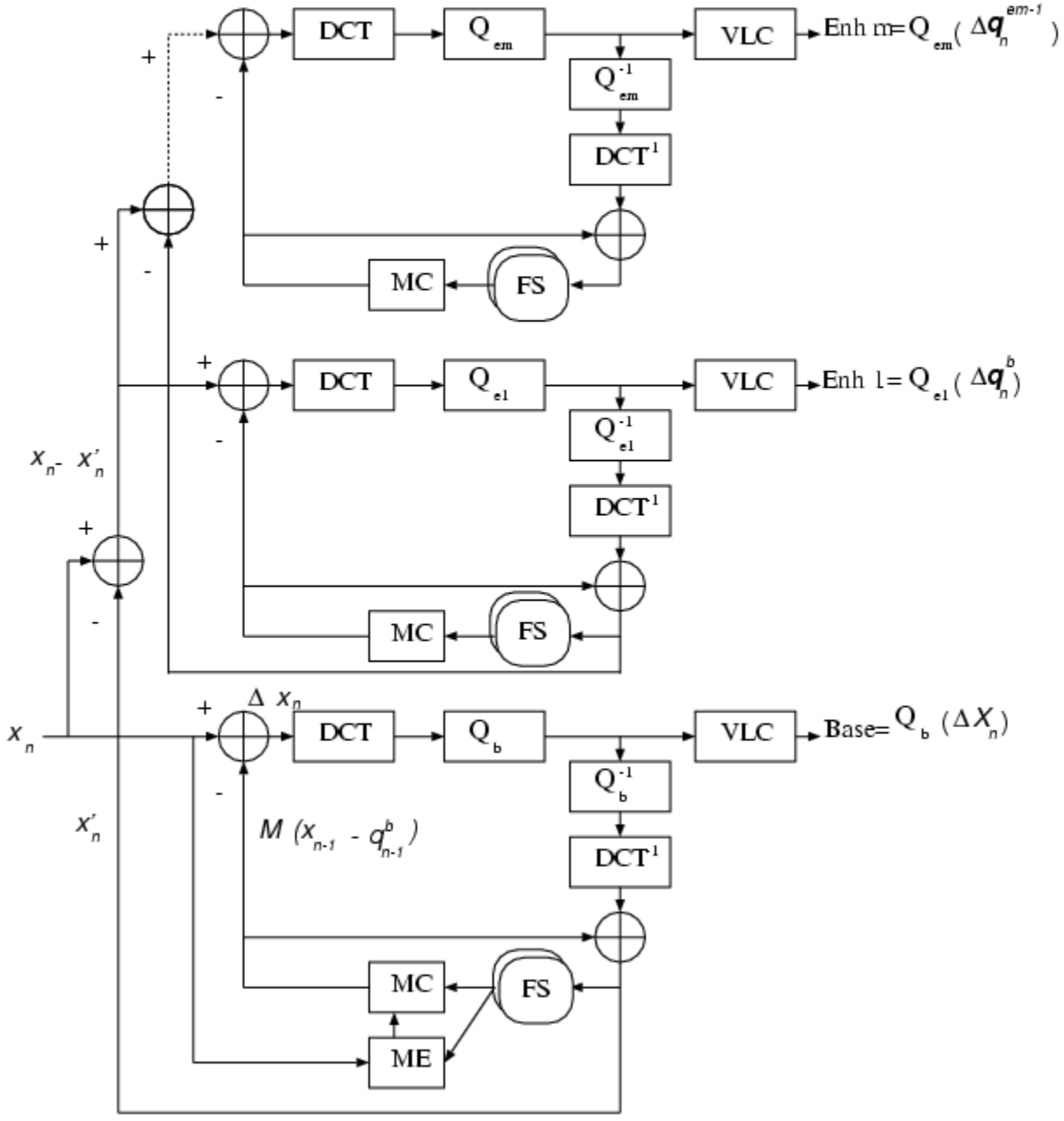


Figure 4. Block diagram of *multi-loop* SNR encoder

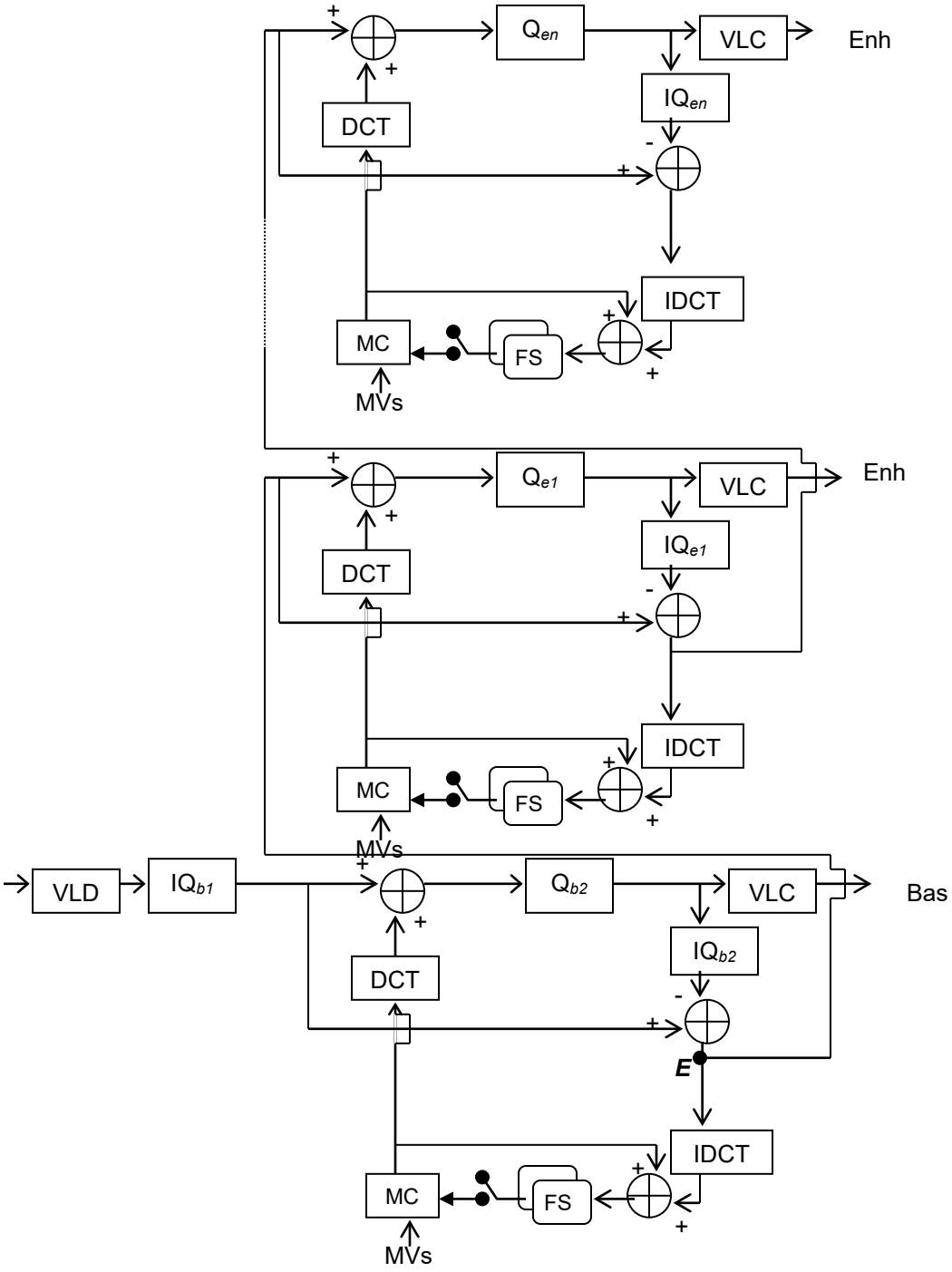


FIGURE 5. Drift Compensated Multilayer (DCM) transcoder based on the multi-loop SNR scalability.

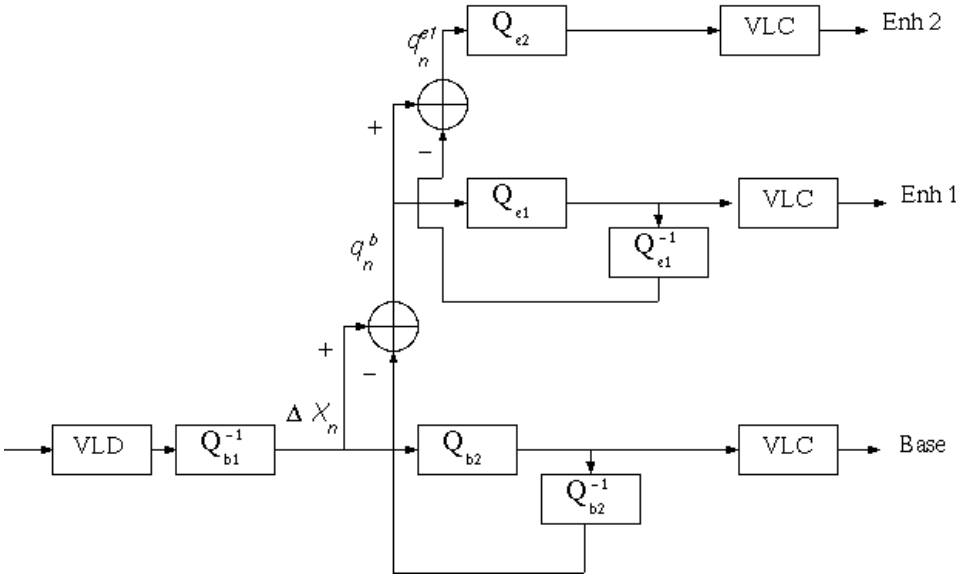


FIGURE 6. Block diagram of a Non-Drift Compensated Multilayer (NDCM) transcoder

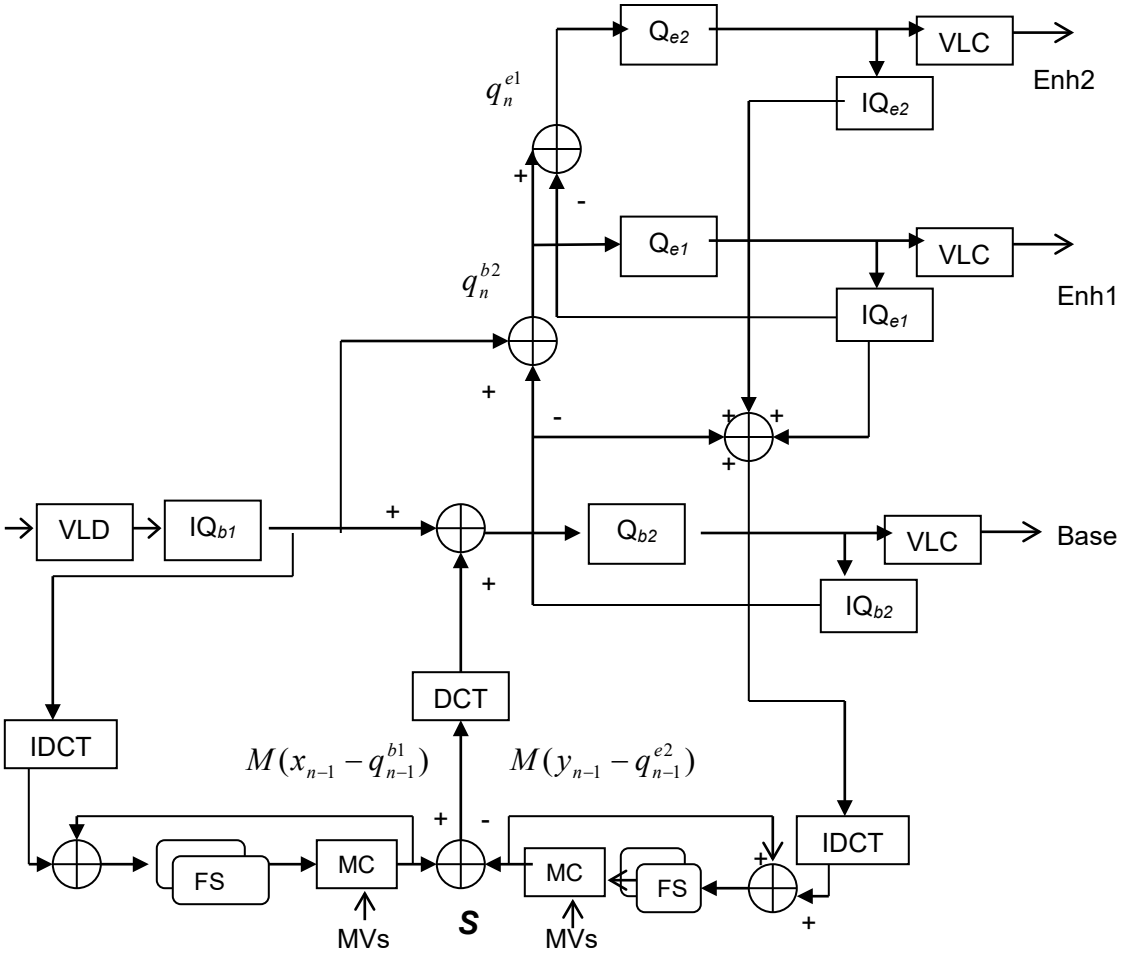
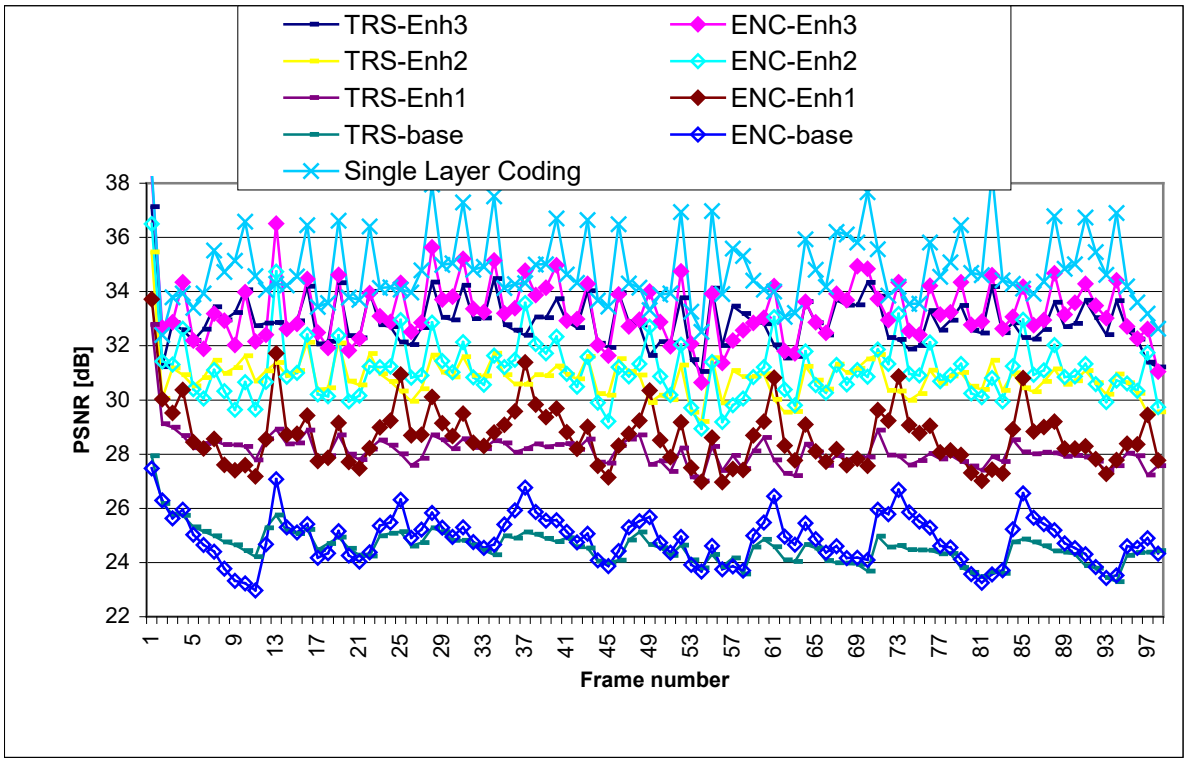
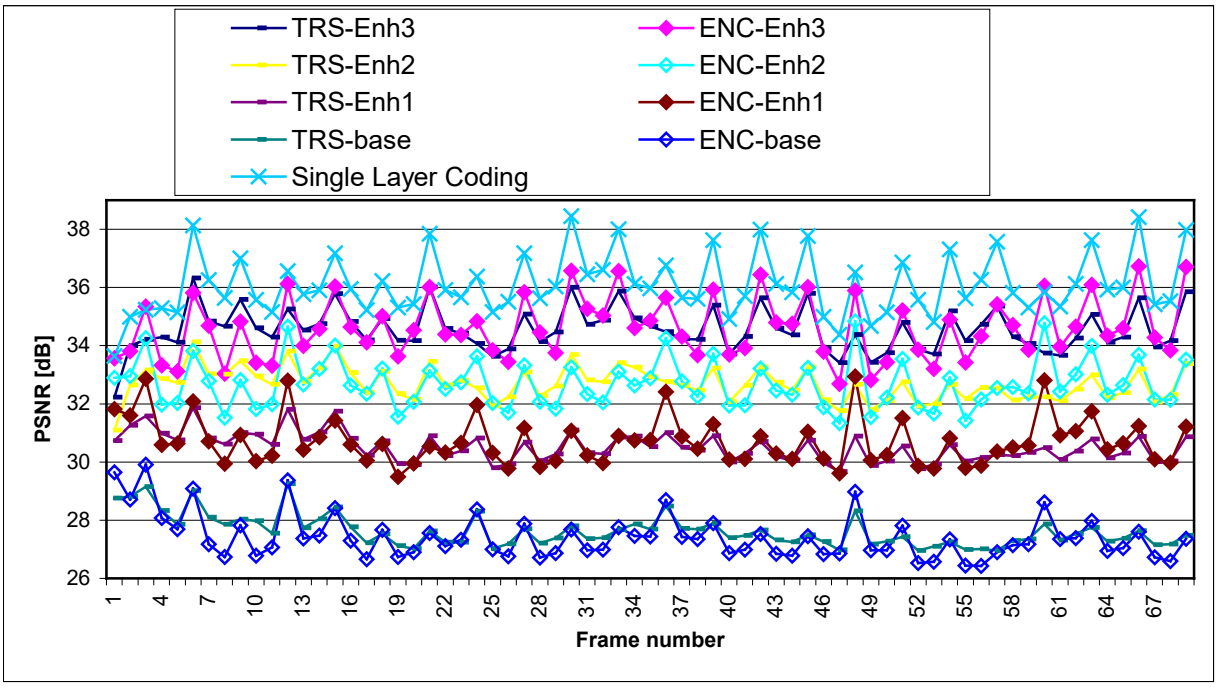


FIGURE 7. Cascade of a decoder and a closed-loop SNR encoder.

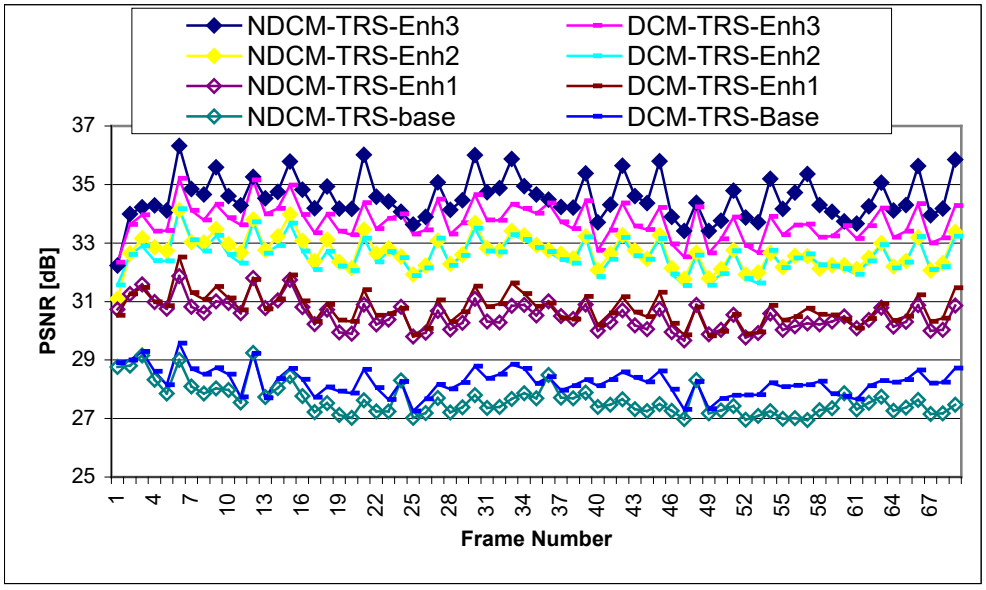


(a)

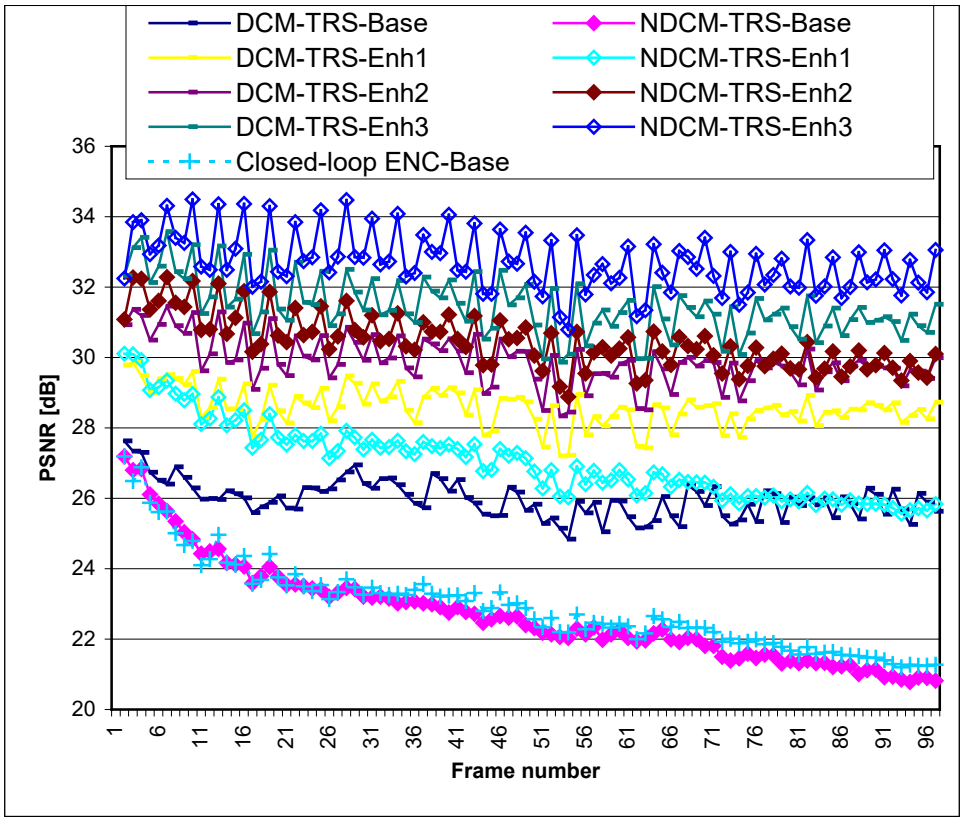


(b)

FIGURE 8. NDCM transcoding (Fig 6) vs. Closed-loop SNR coding (Fig 3) and single-layer coding. (a) Football sequence (b). Flower sequence.



(a)



(b)

FIGURE 9. Multilayering comparison between the NDCM and the DCM transcoders. SIF (25Hz) sequences, encoded at 4Mbit/s, transcoded into 4 layers 1Mbit/s each. (a) Football sequence with GoP structure of (N=12, M=3) (b) Flower sequence with GoP structure of (N= ∞ , M=3).

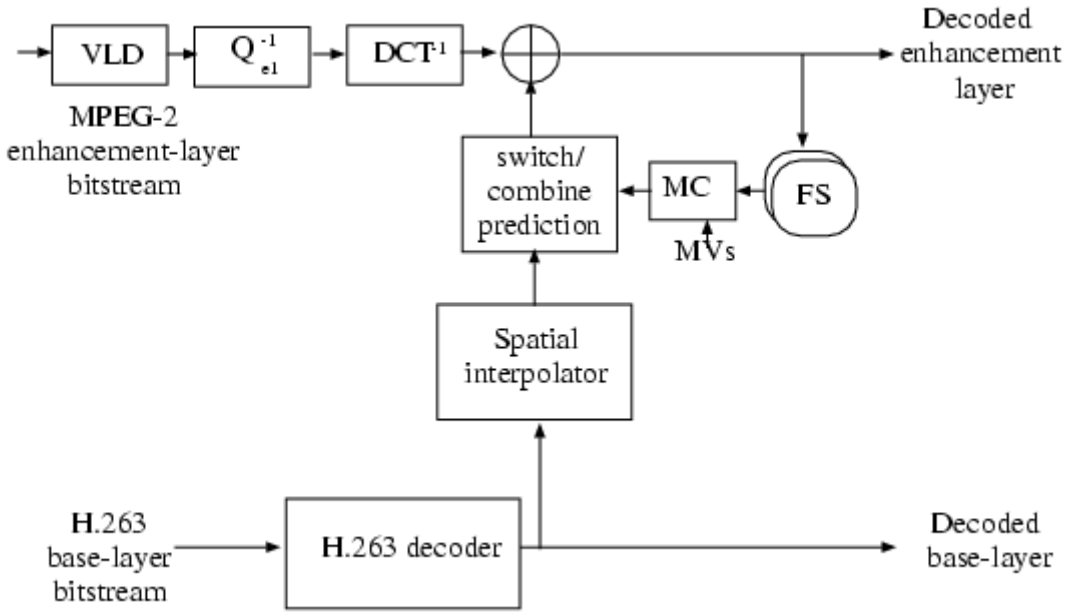


FIGURE 10. MPEG-2 spatial decoder with different encoding formats.

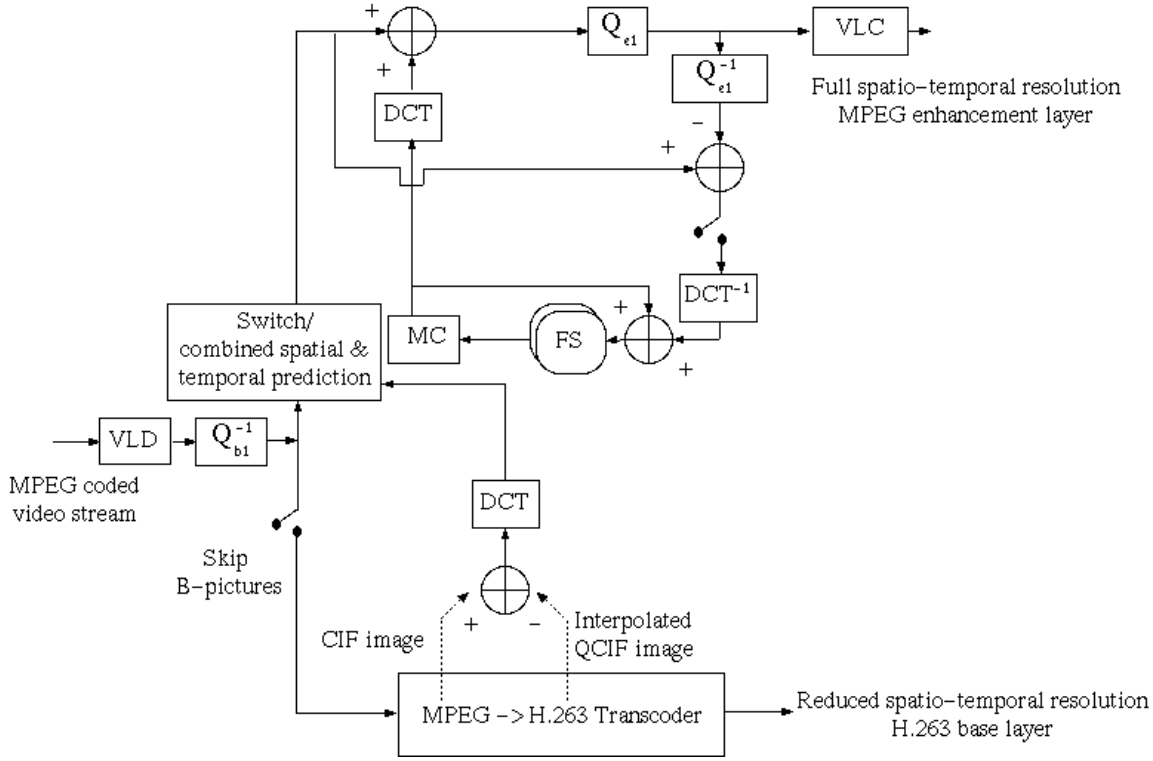


FIGURE 11. Spatio-temporal scalability transcoder with different encoding formats.

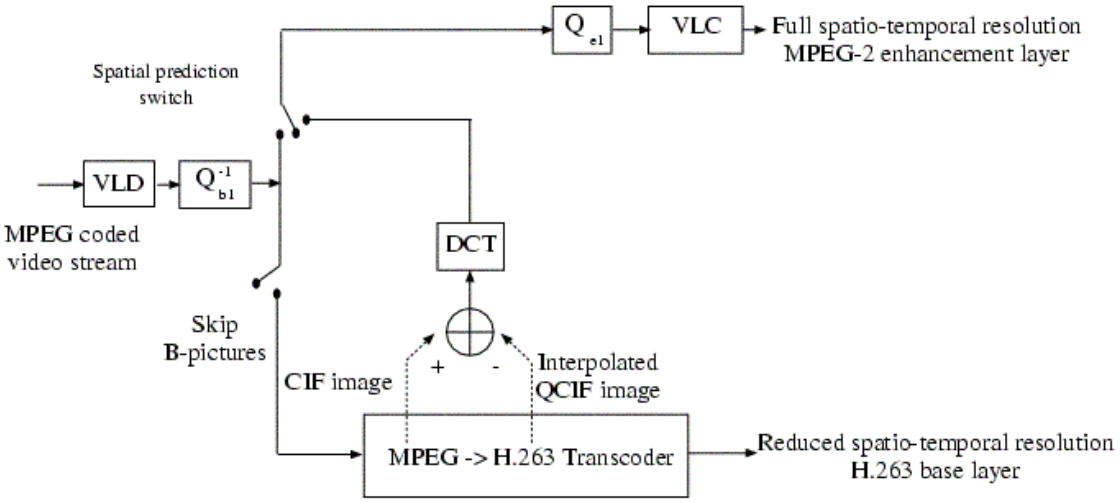
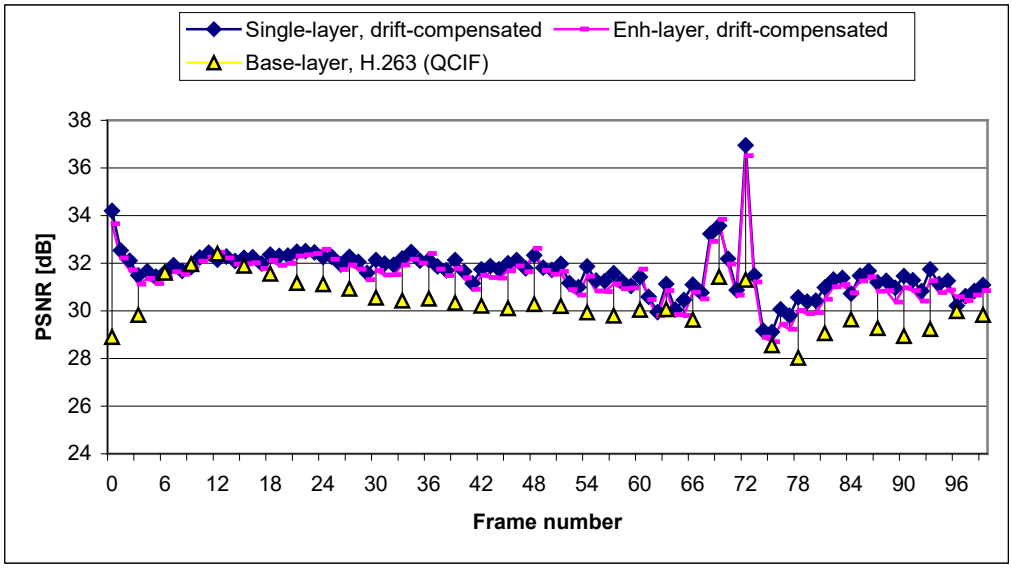
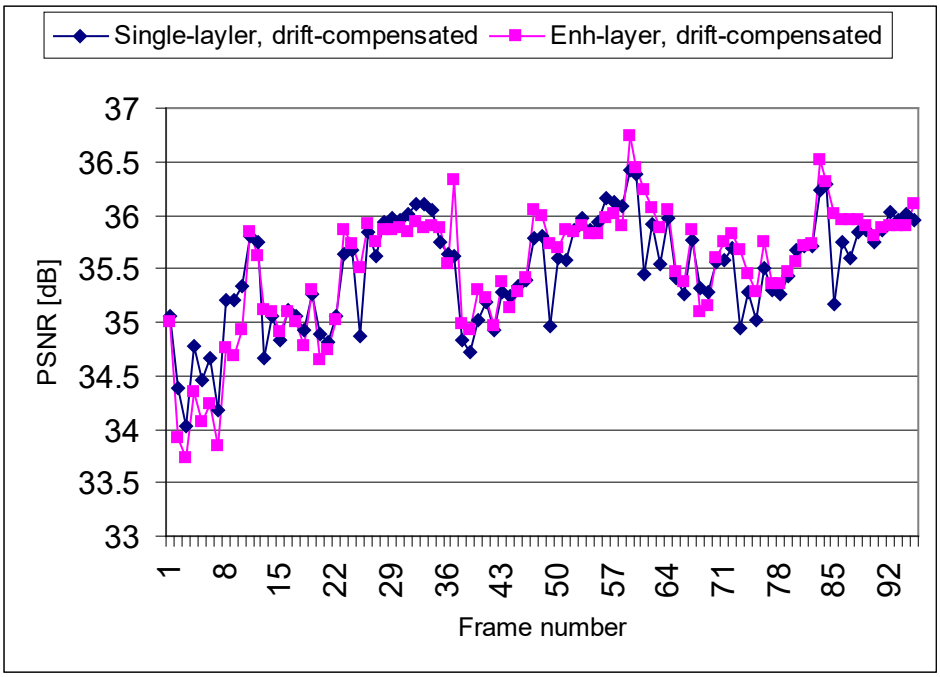


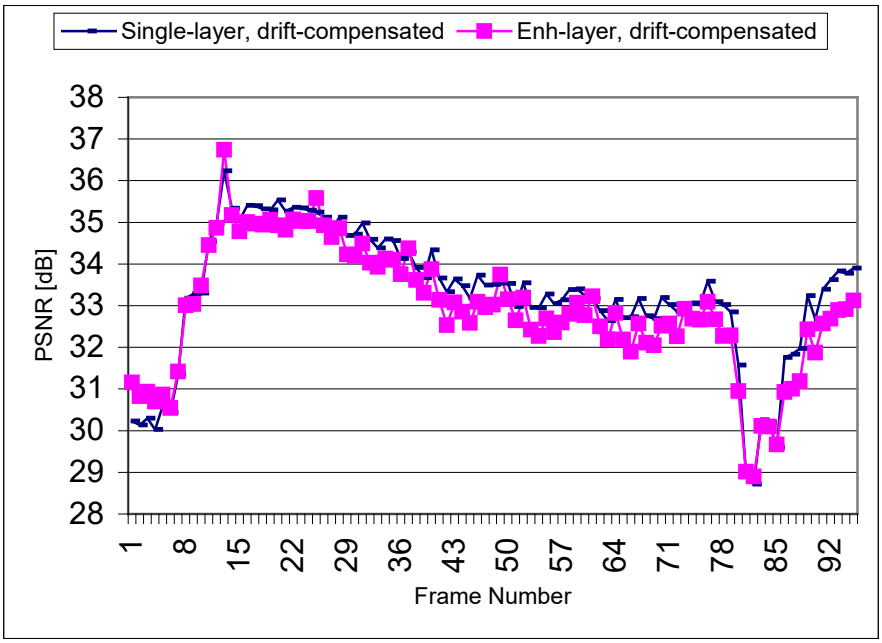
FIGURE 12. A simplified spatio-temporal scalability transcoder.



(a)



(b)



(c)

FIGURE 13. Spatio-temporal scalability transcoding versus single layer transcoding. (a) Coastguard (b) Salesman (C) Table-Tennis.

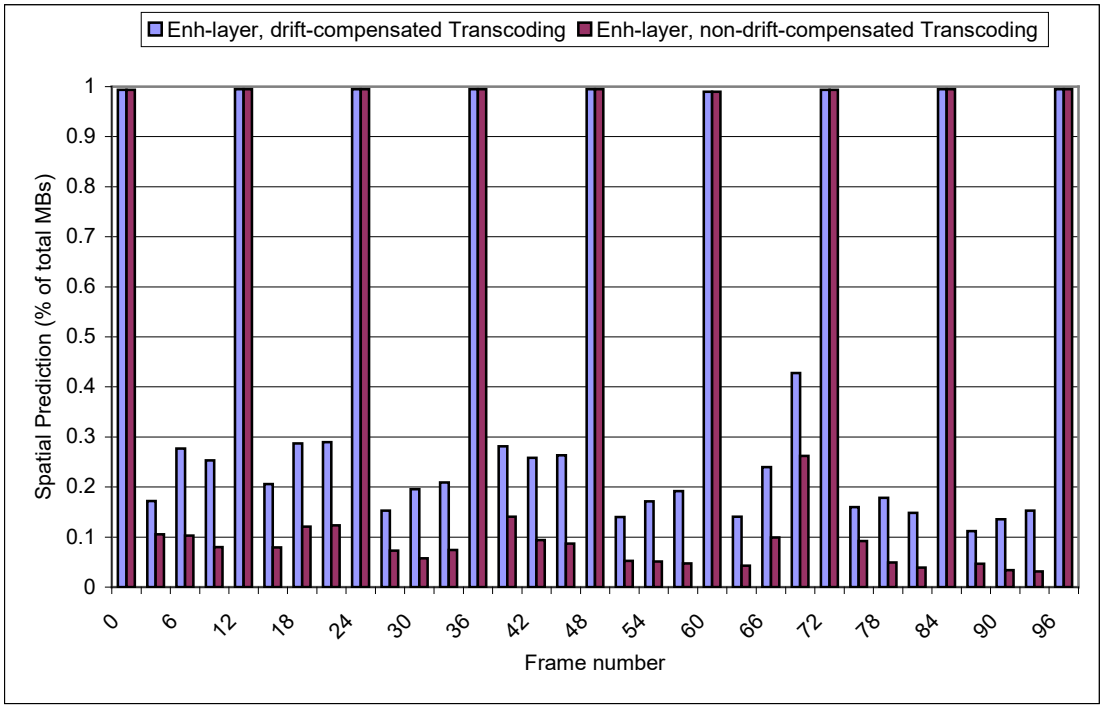
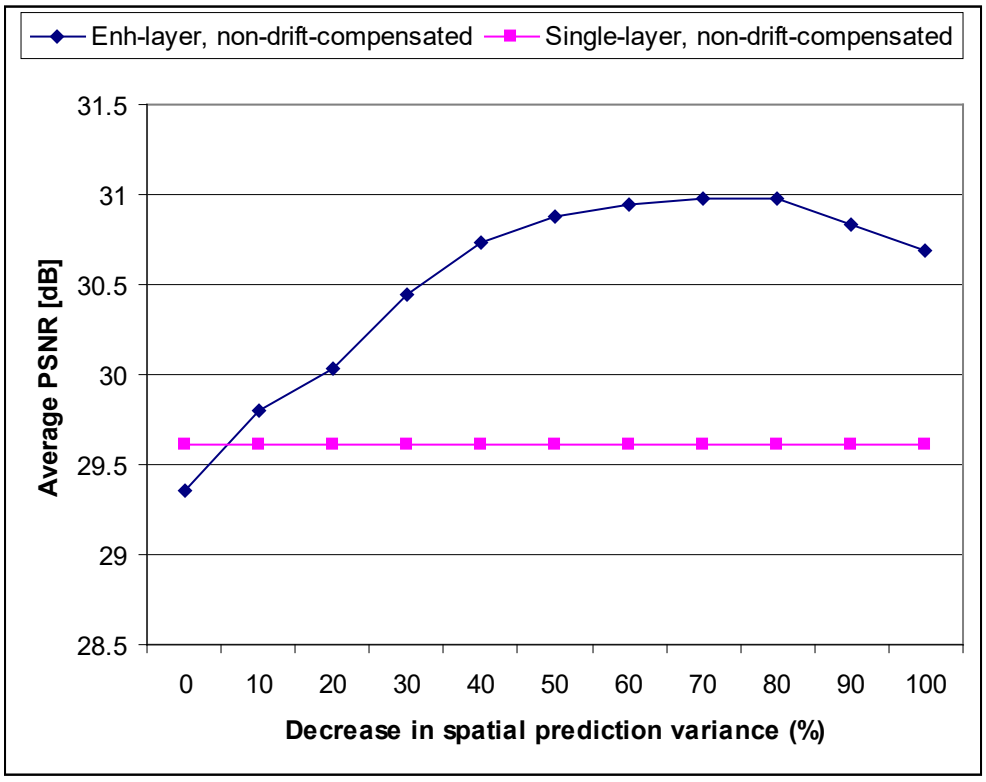
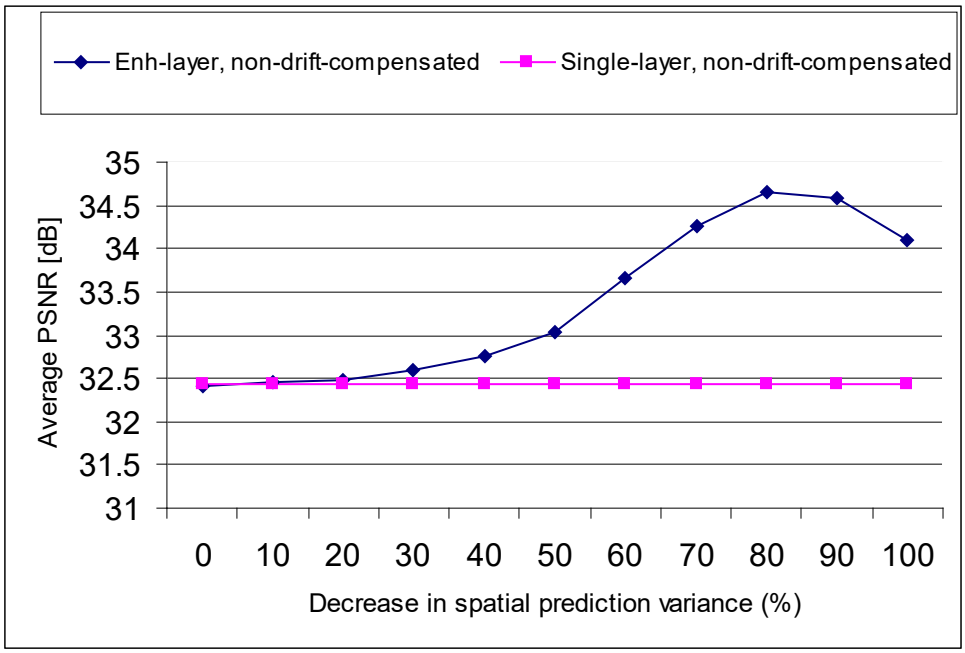


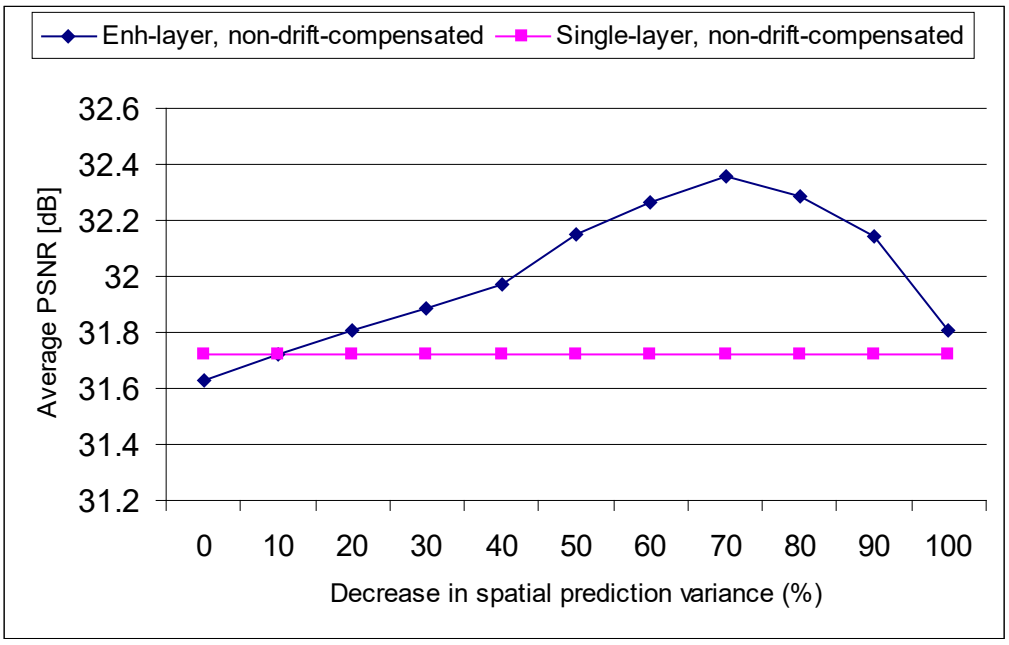
FIGURE 14. Percentage of spatially predicted MBs in I and P pictures. Average results for COASTGUARD and SALESMAN sequences.



(a)



(b)



(c)

FIGURE 15. Effect of increasing the spatial prediction on the overall quality for non-drift compensated transcoding of the enhancement layer (a) Coastguard (b)Salesman (c)Table-Tennis

Appendix A

Architecture verification of MPEG-2 SNR encoders.

1. Closed-loop SNR encoding

In the closed-loop SNR encoder of Figure 3, since the quantizer of a base layer Q_b generates the quantization error $\boldsymbol{q}_n^b$, it follows that the de-quantization of the prediction error ΔX_n results in $\Delta X_n - \boldsymbol{q}_n^b$ where the bold-font and the uppercase '*X*' indicate the DCT domain, the superscripts and the subscripts refer to the source of the quantization error and the underlying picture number respectively[1]. Likewise, the quantizer of the first enhancement layer $Q_{e1;}$ $Q_{e1} < Q_b$ generates the quantization error $\boldsymbol{q}_n^{e1}$; $\boldsymbol{q}_n^{e1} \le \boldsymbol{q}_n^b$ such that the de-quantization of the encoded quantization error of the base layer $\boldsymbol{q}_n^b$ results in $\boldsymbol{q}_n^b - \boldsymbol{q}_n^{e1}$. In general, $\boldsymbol{q}_n^{e_{m-1}}$ is quantized with the next enhancement layer quantizer $Q_{em;}$ $Q_{em} < Q_{em-1}$ such that the de-quantization results in $\boldsymbol{q}_n^{e_{m-1}} - \boldsymbol{q}_n^{e_m}$. Lastly, the de-quantization results of all the layers are summed and fed back to the *MC*-loop of the base layer. In this case the input to the *MC*-loop of the base layer is given by:

$$\begin{aligned} input\ to\ base\ MC-loop &= (\Delta X_n - \boldsymbol{q}_n^b) + (\boldsymbol{q}_n^b - \boldsymbol{q}_n^{e1}) + \ldots + (\boldsymbol{q}_n^{e_{m-1}} - \boldsymbol{q}_n^{e_m}) \\ &= (\Delta X_n - \boldsymbol{q}_n^{e_m}) \end{aligned} \qquad (1)$$

1.1 Architecture verification

For clarity, the following discussion assumes an output of a base and one SNR layer that can be generalized for up to the maximum allowable number of enhancement layers in the MPEG-2 standard defined as 2^4-1.

a. *Combined decoding of the base and the enhancement layers*

[1] The analysis and formalization to follow assumes a two dimensional vector of 8x8 defined for either a block of pixels '*x*' or block of DCT coefficients '*X*'.

Since X_0-$\boldsymbol{q}_0^b$ denotes the de-quantized coefficients of the first I-picture in the base layer and $\boldsymbol{q}_0^b$-$\boldsymbol{q}_0^{e1}$ denotes the de-quantized coefficients of the first enhancement layer. Decoding the two layers up to the DCT coefficients, summing and inverse transforming them, as shown in the decoder of Figure 2, results in the decoded picture d_0 given by:

$$d_0 = x_0 - q_0^b + (q_0^b - q_0^{e1}) = x_0 - q_0^{e1} \quad (2)$$

Hence, the difference between the original picture x_0 and the decoded one d_0 is due to the quantization error of the enhancement layer $q_0^{e1}; q_0^{e1} < q_0^b$.

The decoded d_0 is then motion compensated for the reconstruction of the next picture, or in this case the first P-picture x_1. This results in $M(x_0 - q_0^{e1})$ where *M(.)* denotes the motion compensation process.

Similarly using the SNR decoder of Figure2, the two layers of the first P-picture are decoded up to the DCT coefficients, summed and inverse transformed. This results in $\Delta x_1 - q_1^{e1}$ which is the pixel domain representation of the de-quantized prediction error. This input is then added to the previous motion compensated picture to form the decoded picture d_1:

$$d_1 = \Delta x_1 - q_1^{e1} + M(x_0 - q_0^{e1}) \quad (3)$$

Since the dequantized coefficients of the enhancement layers of the closed-loop encoder are added to the *MC*-loop of the base layer it follows that the prediction error signal Δx_1 can be written as follows:

$$\Delta x_1 = x_1 - M(x_0 - q_0^{e1}) \quad (4)$$

That is, the prediction error is the result of subtracting the raw picture x_1 from the previous dequantized and motion compensated picture x_0.

By approximating the motion compensation to a linear operation[2] as reported in [16,17], and substituting Equation 4 in 3 the final decoded picture becomes:

[2] Note that due to the integer truncation caused by sub-pixel accurate MVs, the *MC* process is in general not a linear operation. However *MC* can be approximated to a linear operation by affording some negligible arithmetic inaccuracy [16,17].

$$d_1 = x_1 - q_1^{e1} \quad (5)$$

More specifically, the finer the quantization error of the first enhancement layer, the higher is the resemblance between the decoded picture and the original one. In general, when a decoded picture is motion compensated for the reconstruction of the next incoming one, the output of the decoder's *MC*-loop i.e. $M(x_{n-1} - q_{n-1}^{e1})$ will be identical to the output of the encoder's *MC*-loop as shown in Figure 3. Therefore, the decoded picture d_n is said to be correctly decoded and shall not cause a *MC*-loop mismatch in the reconstruction of subsequent pictures.

b. Stand alone decoding of the base layer

At the decoder, inverse transforming and de-quantizing the base layer of first P-picture (rather than decoding the base and the enhancement layer) results in $\Delta x_1 - q_1^b$ rather than $\Delta x_1 - q_1^{e1}$. This prediction error is then added to the previous de-quantized motion-compensated picture, which in this case was also decoded without the enhancement layer. The decoded P-picture is given by:

$$d_1 = \Delta x_1 - q_1^b + M(x_0 - q_0^b) \quad (6)$$

substituting Δx_1 by its value at the encoder as given in Equation 4 we get:

$$d_1 = x_1 - M(x_0 - q_0^{e1}) - q_1^b + M(x_0 - q_0^b) \quad (7)$$

Again note that the de-quantized output of the enhancement layer $q_0^b - q_0^{e1}$ is added to the *MC*-loop of the encoder's base layer but not to that of the decoder. The final decoded pictures becomes:

$$d_1 = x_1 - q_1^b - M(q_0^b - q_0^{e1}) \quad (8)$$

Equation 8 states that the difference between the decoded picture and the original one x_1 is not just due to the quantization error of the base layer but also due to the erroneous term $M(q_0^b - q_0^{e1})$. More specifically, while the output of the enhancement layer is added to the base layer *MC*-loop of the closed-loop encoder, this enhancement layer bitstream is absent at the decoder and therefore do not contribute to its *MC*-loop. Consequently, the motion compensated picture of the encoder's *MC*-loop $M(x_0 - q_0^{e1})$ mismatches that of

the decoder's *MC*-loop $M(x_0 - q_0^b)$ causing a *MC*-loop mismatch that resulted in the erroneous term of Equation 8.

2. Multi-loop SNR encoding

2.1 Verification of the architecture

a. Stand alone decoding of the base layer

In this architecture the base layer is coded without feedback to its motion compensation loop from the enhancement layers as shown in Figure 4. The loose-coupling between the video layers implies that the base layer is correctly decoded by the SNR decoder regardless of the existence of the enhancement layers. This is identical to the single layer decoding hence no further verifications are given.

b. Combined decoding of the base and the enhancement layers

As pointed-out earlier, the enhancement layer of the multi-loop SNR encoder is encoded by means of motion compensation. Hence in general, the de-quantized *prediction error* coefficients of a P-picture in the first enhancement layer are given by $\Delta \boldsymbol{q}_n^b - \boldsymbol{q}_n^{e1}$ rather than $\boldsymbol{q}_n^b - \boldsymbol{q}_n^{e1}$. Consequently, a decoded P-picture can be written as:

$$d_n = \Delta x_n - q_n^b + (\Delta q_n^b - q_n^{e1}) + M(x_{n-1} - q_{n-1}^{e1}) \qquad (9)$$

Where Δq_n^b is generated by subtracting the input signal q_n^b from the previous de-quantized motion compensated one; $M(q_{n-1}^b - q_{n-1}^{e1})$ given by:

$$\Delta q_n^b = q_n^b - M(q_{n-1}^b - q_{n-1}^{e1}) \qquad (10)$$

Similarly, substituting Δx_n by $x_n - M(x_{n-1} - q_{n-1}^b)$ we get:

$$
\begin{aligned}
d_n &= x_n - M(x_{n-1} - q_{n-1}^{b}) - q_n^{b} + q_n^{b} - M(q_{n-1}^{b} - q_{n-1}^{e1}) - q_n^{e1} + M(x_{n-1} - q_{n-1}^{e1}) \\
&= x_n - M(x_{n-1}) + M(q_{n-1}^{b}) - M(q_{n-1}^{b}) + M(q_{n-1}^{e1}) - q_n^{e1} + M(x_{n-1}) - M(q_{n-1}^{e1})
\end{aligned}
\tag{11}
$$

Hence, the final decoded picture is reconstructed correctly to $x_n - q_n^{e1}$.

Therefore, both the base and enhancement layers of the multi-loop encoder are decoded correctly without any picture drift.

Appendix B

Verification of the multi-loop Transcoding architecture

The notations and assumptions introduced in Appendix A are reused throughout the following verifications. One difference applies which is due to the irrecoverable re-quantization error in video transcoding. Hence in the verifications to follow two sources of quantization errors are considered; the original encoder's quantizer error referred to as q_n^{b1} and the transcoder's one q_n^{b2}.

Referring to the multilayer transcoder of Figure 5, the output of the base layer for the first P-picture is given by

$$B_1 = Q_{b2}(DCT[\Delta x_1 - q_1^{b1} + M(q_0^{b2})]) \quad (12)$$

Again, the superscripts and the subscripts refer to the source of the quantization error and the underlying picture number respectively. Additionally, the previously decoded signal in the term $M(q_0^{b2})$ is the motion compensation of the quantization error resulting from transcoding the first I-picture. As shown in the transcoder of Figure 5, after requantizing the I-picture in the base layer, the dequantized coefficients are subtracted from the incoming ones resulting in the quantization error q_0^{b2}. By reusing the incoming motion vectors this quantization error is then motion compensated to generate the signal $M(q_0^{b2})$. This motion compensated signal is then added to the incoming prediction error $DCT[\Delta x_1 - q_1^{b1}]$ belonging to the first P-picture. The resultant signal is then quantized using the coarser base quantizer Q_{b2}.

Similarly, referring to the same figure, the input to the first enhancement layer indicated by *"E"* in the block diagram of the base layer is q_0^{b2} for the first I-picture. However, for the following P-picture the input signal to the enhancement layer is given by:

$$
\begin{aligned}
E_1 &= DCT[\Delta x_1 - q_1^{b1}] - (DCT[\Delta x_1 - q_1^{b1} + M(q_0^{b2})] - q_1^{b2}) \\
&= DCT[q_1^{b2} - M(q_0^{b2})] \\
&= DCT[\Delta q_1^{b2} - M(q_0^{e1})] \quad where \quad \Delta q_1^{b2} = q_1^{b2} - M(q_0^{b2} - q_1^{e})
\end{aligned}
\tag{13}
$$

Hence the output of the enhancement layer for the first P-picture is given by:

$$
\begin{aligned}
Enh_1^1 &= Q_{e1}(DCT[\Delta q_1^{b2} - M(q_0^{e1}) + M(q_0^{e1})]) \\
&= Q_{e1}(DCT[\Delta q_1^{b2}])
\end{aligned}
\tag{14}
$$

By decoding the two layers up to the DCT coefficients, summing and inverse transforming them as shown in the decoder of Figure 3, the first decoded I-picture is given by:

$$
d_n = x_0 - q_0^{b1} - q_0^{b2} + q_0^{b2} - q_n^{e1} = x_0 - q_0^{b1} - q_n^{e1} \tag{15}
$$

Likewise, for the first P-picture, the two layers are decoded up to the DCT coefficients, summed and inverse transformed. According to equation 12 and 14 above this results in the following signal $\Delta x_1 - q_1^{b1} + M(q_0^{b2}) - q_1^{b2} + \Delta q_1^{b2} - q_1^{e1}$.

The above signal is then added to the motion compensation of the first I-picture which results in the following decoded picture:

$$
d_1 = \Delta x_1 - q_1^{b1} + M(q_0^{b2}) - q_1^{b2} + \Delta q_1^{b2} - q_1^{e1} + M(x_0 - q_0^{b1} - q_0^{e1}) \tag{16}
$$

As shown in Appendix A, substituting Δq_1^{b2} by $q_1^{b2} - M(q_0^{b2} - q_1^{e})$ and Δx_1 by $x_1 - M(x_0 - q_0^{b1})$. The final decoded picture becomes:

$$
d_1 = x_1 - q_1^{b1} - q_1^{e} \tag{17}
$$

Hence the difference between the reconstructed picture d_1 and the original one x_1 is due to both the irrecoverable quantization error of the original encoder and the quantization error of the transcoder's enhancement layer.