Video Aggregation : An Integrated Video Compression and Multiplexing Scheme for Broadband Networks

Chi-yin Tse and Soung C. Liew
Department of Information Engineering, The Chinese University of Hong Kong

***Abstract*— This paper investigates video aggregation, a concept that integrates compression and multiplexing of video information. We focus on the transmission of a group of video sessions as a bundle, the practical examples of which include entertainment-video broadcast and video-on-demand. The shortcomings of the approaches that separate the processes of compression and multiplexed are explained from the viewpoints of image quality, bandwidth usage, network management and operation. We argue that it is better to perform compression and multiplexing together before the bundle of video traffic enters the network. This paper presents preliminary experimental results which indicate that video aggregation can provide better image quality for a given bandwidth than traditional statistical multiplexing.**

I. Introduction

Future broadband integrated services networks based on the ATM (Asynchronous Transfer Mode) technology are expected to carry information from a large variety of different services and applications. However, video traffic is likely to dominate, thanks to the bandwidth-hungry nature of images. It is therefore important to understand how video traffic might best be multiplexed, transported, and switched.

In ATM networks, data are packetized into fixed-length cells of 53 bytes. Cells are routed in the network independently based on the routing information contained in their 5-byte headers [1]. These cells may be discarded inside the network when traffic congestion occurs.

To reduce the bandwidth needed, video is almost always compressed before transmission. The Moving Picture Experts Group (MPEG) coding scheme [2] has been developed and evolved as a standard of video compression. Since data have been highly compressed, cell loss during transmission of MPEG coded video may cause serious degradation of image quality. Low-cell-loss-rate network operation, or schemes that facilitate such operation, is therefore essential.

This paper focuses on the scenario where information from a group of video sessions are to be delivered as a bundle to one more destinations. We argue that compression and multiplexing of video streams in such a scenario should occur together before packetization (i.e., before the ATM layer[1]). To distinguish this from statistical multiplexing of cells, we call this *video aggregation*. We argue and demonstrate the advantages of aggregation from the viewpoints of image quality, bandwidth usage, network management and operation.

Application areas of aggregation include video broadcast and video-on-demand. Video programs are transported as a bundle from the video server directly to the subscribers in the former, and to a distribution node close to the subscribers in the latter [3]. Aggregation may also find use in the transport of long-distance videophone data: video streams from various subscribers targeted for a common remote area may be aggregated at a local central office before being delivered as a bundle to the remote central office serving the area.

II. Motivations and Basic Concepts of Video Aggregation

Let us first consider a video stream before moving on to a video bundle. Compression methods of a video stream can be divided into two classes: variable bit-rate (VBR) compression and constant bit-rate (CBR) compression. For a given video stream, its bandwidth requirement may vary over time just to maintain a constant image quality, thanks to the variation of the scene contents of the underlying video sequence. In VBR compression, the output bit rate of the encoder varies over time according to the bandwidth requirements of the underlying video sequence. Therefore, the image quality is more or less constant. In CBR compression, the output bit rate of the encoder is forced to be constant. The image quality varies over time since scenes that intrinsically demand high bandwidths may have their bandwidths cut down to maintain the constant output bit rate. Compression schemes that lie somewhere between the two extremes are also possible. In general, in the consideration of compression, there is a kind of "uncertainty principle" which consists of the tradeoff between the variations of bit rate and image quality.

CBR and VBR transport, as distinct from compression, refers to using constant-bit-rate and variable bit-rate channels, respectively, for the transport of data, CBR transport has many advantages from the network viewpoint. Since the data rate is constant, bandwidth allocation and charging for network usage are simple.

It is also straightforward for the network to multiplex several CBR channels onto a common communication channel and guarantee the delivery of cells since cells arrive at predictable rates.

It is natural to use CBR transport for CBR-compressed data. Similarly, VBR compression followed by VBR transport is a natural combination. In the second case, however, it is difficult to multiplex VBR video streams while guaranteeing the delivery of all cells, since the multiplexed streams may all output a large number of bits together simultaneously. Generally, more than the average bandwidth needs to be allocated to a VBR stream to maintain a small cell-loss probability. Even then, absolute cell-delivery guarantee is not possible unless the peak bandwidth is allocated, in which case the delivery of the VBR stream will be much more expensive than the corresponding CBR stream. For public networks, the fact that cells may be dropped due to interference from other streams also complicate the charging problem and the contractual agreement between the network operator and user.

The other combination that makes sense is the use of CBR transport for VBR-compressed data. The VBR data from the output of the VBR encoder is fed to a smoothing buffer, which then forwards data at a constant rate to the network. However, if the buffer size is not large enough then data may be dropped due to buffer overflow. Furthermore, data also incur delay jitters in the buffer in addition to those in the network.

One general issue is how to achieve the advantages of VBR compression (which offers relatively constant image quality) and advantages of CBR transport (which facilitates simple network operation). It turns out that this is possible when several video streams are to be transported as a bundle. A common CBR channel can be used to transport the VBR-compressed streams as a whole. In other words, as a group, the video bundle is CBR, but individually, the video streams are VBR. The contract between the network and the user is simple: the network is required to guarantee the delivery of all cells so long as total data rate of the streams does not exceed the CBR-channel bandwidth.

To combine the video streams on to the common CBR channel, the straightforward method is to first packetize the output data of the VBR encoders into cells and then statistically multiplex the cells [4, 5, 6]. If not all video streams demand high bit rates simultaneously, we may smooth the combined traffic statistically. A problem, however, is that cells might still be dropped by the user at the edge when the VBR streams all ouput high bit rates simultaneously.

A two-layer video coding and transport strategy has been investigated by many researchers for minimizing the quality degradation due to cell loss [7, 8, 9, 10, 11]. In this approach, VBR-coded data of a video stream are divided into two groups: the base layer contains data that correspond to basic-quality images, and the second layer contains image-enhancement data. Two-layer coding can be in the video bundle scenario as follows. Each video source first partitions its output traffic into two streams: the guaranteed stream (GS) is made up of the base-layer data and the enhancement stream (ES) is made up of the second-layer data. The GS's from the video sources are first transmitted independently guaranteed and they use up certain amount of the bandwidth of the CBR channel. The remaining bandwidth is used in multiplexing of the ES's. Cells of the ES's may be dropped when the common CBR channel does not have enough bandwidth to accommodate all the data.

The shortcoming of the two-layer approach is that there is no distinguishing between the relative importance of data within an ES and among the separate ES's (the implications will be described in more detail in Section III B). In fact, given that it is not necessary to multiplex data at the cell level at the edge of the network, we might as well do it before the data have been packetized. This is because we then know the relative importance of the data down to the last detail and selectively drop data that are least significant. In this way, the two facts that 1) within a video stream, not all data are equally important, and 2) between the video streams, some streams may require more bandwidth than the other streams at a particular moment in time, can be fully exploited to achieve 1) better and smoother image quality for the frames of each video stream , and 2) fairness of image quality among the video sequences. This is the basic observation that gives rise to the concept of video aggregation.

In video aggregation, attempts are made so that 1) sum of the coded bit rates of the video sequences is almost equal to (but not larger than) that reserved by the CBR channel, and 2) a more or less equal image quality among the video streams. To do this we have to integrate multiplexing with compression: compression is performed such that the aggregated output bit rate is constant and that the separate video streams have roughly the same image quality according to some signal-to-noise or distortion metric. Video aggregation is described in an abstract manner below as a lossy compression process that is applied after a preliminary compression process.

Description of Video Aggregation:
In many video compression schemes, the output data can be divided into segments. Each segment has a certain number of bits, some of which can be dropped, if needed, for the price of image-quality degradation. As-

sociated with each segment is a function relating the number of bits retained and the corresponding image quality. Within each segment, bits can be ordered according to their significance so that those of lower significance will be dropped first when necessary.

As an illustration, in MPEG coding (see next section) the segments could be "blocks" and the bits could be codewords representing the nonzero frequency components in the blocks. Bits in a block can be ordered according to frequency because the codewords of low frequencies are generally more significant to image quality.

In aggregation, a number of segments from each video source is collected. Let N be the total number of segments collected from all sources. Let $B_i(D)$ be the number of bits in segment i that must be retained in order to maintain a distortion level of D. The goal is to find a distortion level D' such that

$$B_1(D') + B_2(D') + \cdots + B_N(D') = B_t$$

That is, when B_t is insufficient to transport all the bits of all aggregated segments, bits are dropped until the above equality can be achieved. Notice that the objective is that all segments achieve the same distortion D' as a result.

In practice, it may not be possible to achieve absolute equality of distortion levels because of the discrete nature of bits or groups of bits (e.g., codewords) that are dropped. In this case, the aim is to transport no more than B_t bits and minimize the difference between distortion levels of any two segments.

III. MPEG Video Aggregation System

We now explain in more detail how the concept of aggregation might be applied with respect to the MPEG coding standard. As preliminaries, let us first review the basics of MPEG coding, as well as the problems of cell-level multiplexing from the viewpoint of image quality.

A. MPEG Coding

The schematic of an MPEG coder is shown in Fig. 1 [7]. In MPEG coding standard [2], spatial information of a frame is partitioned into four layers: frame, slice, macroblock and block. A frame is the basic unit of display, and is further divided into slices. A slice is a sequence of macroblocks (MB). A 16×16 (16 pixels by 16 pixels) macroblock (MB) is the unit for motion compensation, and it consists of 8×8 blocks. Discrete cosine transform (DCT) is performed on each 8×8 blocks. For color video, an MB consists of four 8×8 luminance blocks and two 8×8 chrominance blocks. For each frame, there are three choices of coding algorithms: Intraframe, interframe and interpolative coding.

Intraframe coded frames (I) are coded independently. The whole I frame undergoes 8×8 block-based DCT without referring to other frames. The DCT coefficients are then quantized. The DC coefficients of individual blocks are coded differentially within a slice. For variable-length coding (VLC), each non-zero AC component is first grouped with the runlength of preceding zero components (in zig-zag order, see Fig.2), and then assigned a codeword from a Huffman table.

For interframe coded frames (P), temporal redundancy is first reduced by causal MB-based motion compensation, with respect to the preceding I or P frames stored in the Frame Storage. If the motion estimation (ME) error for a MB is less than a threshold (i.e., there is enough redundancy that interframe coding is worthwhile), then the motion vector (MV) will be differentially and then VLC coded, while the ME error will undergo DCT, coarse quantization and then VLC. Otherwise, that MB will undergo intraframe coding. Interpolative frames (B) are coded in a way similar to coding P frames, however, the motion compensation is bi-directional with respect to both the preceding and following P (or I) frames. For more details about the MPEG standard, please refer to [2, 7, 12].

An MPEG coder is characterized by three parameters : quantization factor q, N, and M. q control the degree of fineness of quantization. $N-1$ is the number of frames coded between successive I frames, while $M-1$ is number of B frames coded between successive P frames. A Group of Frames with $N = 10, M = 3$ is as follows.

IBBPBBPBBP

Data in an MPEG coded video stream are of different importance. The header information, MV's and DC components are obviously very important. Among the DCT AC components, those of lower frequencies are more important than those of higher frequencies for two reasons. First, the energy (i.e., amplitude square) of the DCT AC components tends to decrease along the zig-zag scanning order (i.e., energy compaction) [13]. Second, human vision system is less sensitive to the high frequency signals.

When some data in I and P frames are lost during transmission, the frame contents in the Frame Storages at the coder and decoder become different. Even if no further data is lost, for the following P and B frames, the ME at the coder and decoder will refer to different frame contents as the "baseline" of estimation. Consequently, errors due to data loss of one I or P frame will propagate along the following P and B frames, and this is often referred to as error propagation. The accumulated errors can be cleared by sending an I frame.

B. Shortcomings of the MPEG Video Bundle Scenario with Two-Layer Coding and Cell-Level Multiplexing

In Section II, we have claimed that the shortcoming of the cell-level multiplexing with two-layer coding and transport is that there is no distinguishing between the relative importance of data within an ES and among the separate ES's. Let us now examine its implications for image quality in mode detail.

1. Blocky Effects within a Frame
In multiplexing ES's, the discarding of an ES cells means that those MB's corresponding to this cell (in general, one to ten MB's [8]) will have transmitted their base-layer data only, and hence can provide only basic image qualities. Unless all cells from a frame can be transmitted, the MB's within a frame will have different qualities due to the dropping of some ES cells and the retaining of others. This results in *blocky effects* on the reconstructed image (image appears as clusters).

2. Non-optimal Image Quality within a Frame
There is no prioritization among the second-layer data once they have been packetized into ES cells, even though the second-layer data in the cells may be of different importance. An ES cell is either dropped or transmitted in its entirety. We cannot, say, drop part of an ES cell and part of another ES cell so as to ensure that only the least significant data are dropped. Thus, Optimality is not achieved because some of the dropped data may potentially contribute more to the quality of the reconstructed images than those retained.

3. Fairness of Image Quality among the Video Sequences
Consider the video streams that are multiplexed. When cells must be dropped at the multiplexer, the multiplexer does not have the knowledge of the significance levels of the ES cells to their associated images. It is possible that some images (or portions of an image) suffer more visual degradation than others, even if they incur the same cell-loss rate. The problem of the lack of a measure of the signal degradation due to cell loss is further compounded by the fact that the importance of the cells varies from intraframe to interframe coding. For example, a cell from an I frame may carry 5% of the signal of the reconstructed image; however, one from a P frame may carry 5% of the ME error, which contributes to only 0.5% of the overall image signal. Certainly, dropping a cell from a P frame is more tolerable than dropping one from an I frame. The multiplexer for two-layer transport does not generally distinguish between P-frame and I-frame cells.

C. MPEG Video Aggregation

The goal of MPEG video aggregation is to ensure that all MB's contained in the corresponding spatial unit (slice or frame) from all video sequences provide more or less the same image quality. In our implementation, MPEG video aggregation is slotted into slice periods. [1] In every slice period, data for a slice (which are still in the form of VLC codewords and not yet packetized) is collected from every MPEG video sequence. A number of bits are allocated for all the slices to be aggregated. All the header information, MV's as well as the first β codewords from every 8×8 block are forwarded. This uses up a certain amount of bandwidth. The remaining codewords are then subject to aggregation with the remaining bandwidth B (note that B may change from aggregation period to aggregation period).

There are two reasons why we might want to exempt the first β codewords from the aggregation process. The first reason is that this will reduce the amount of data to be aggregated and hence the complexity of the process. The second reason, which is more subtle, is that in some variations of aggregation systems, it might be advantageous to do so (see Section IV).

At the beginning of the aggregation process, the distortions of all MBs are calculated. The MB which currently provides the lowest image quality is identified. If there are remaining bits, the next codeword from all the 8×8 blocks contained in that MB will be forwarded. This step is repeated until all the allocated bits for that slice period have been used up.

Note that because the codewords for each 8×8 block are arranged with their DCT components in the zigzag order (see Fig. 2), for each block, the codewords discarded during aggregation are of higher frequencies and hence less important.

The signal-to-noise ratio (SNR) is commonly used as an objective measurement for image quality. However, since most of the redundancy in P and B frames has already been removed by motion compensation, the actual signal energy of an MB from P or B frames can be found only after it has been decoded (with respect to the reference frame) back into the spatial domain. Therefore, unless signal energy in each MB is provided by the MPEG encoders, using SNR as the metric for image quality during aggregation is not feasible (unless, of course, the aggregator decodes the MPEG sequences to find out the signal energies of MB's). Alternatively, we may use noise energy as the metric. As the amount of energy carried by a codeword is equal to the amplitude square of the non-zero DCT component contained, the noise energy in an MB is equal to the sum of the energy of the discarded (or not-yet-chosen) codewords.

D. MPEG Video Aggregation System Architecture

We now look at the overall architecture of the MPEG video aggregation system (VAS). An MPEG VAS com-

[1] In general, the unit of aggregation can be smaller or larger than a slice, depending on the processing capability.

prises a group of MPEG video sources, a VAS server and the ATM Adaptation Layer (AAL) [1] (Fig. 3).

Video sequences are coded independently by MPEG coders with high qualities. The coded data are then forwarded to the VAS server (without packetization). The VAS server is responsible for aggregating the video sequences, as well as reassembling the forwarded data block by block after aggregation. If the codewords in the video sequences have been Huffman-coded, the VAS server should also Huffman-decode them first before performing aggregation.

At the AAL, data of the same video sequence are packetized into cells, and cells from all sequences are transported by a single virtual circuit (VC) (or virtual path, VP).

In principle, the allocated number of bits for a slice period can either be fixed or varied. In the first case, the output from the VAS enters the CBR channel of the network directly. Temporal statistical multiplexing (i.e., smoothing of traffic generated at different time instants) is confined in a slice period only. In the second case, the output enters a buffer which in turn outputs data at a constant rate to the network; the allocated number of bits to a slice varies according to the state of the buffer occupancy. The second case allows for smoothing of traffic over a longer time period as compared to the first case at the expense of more complicated operation and additional delay jitters at the buffer. However, because the aggregated traffic should be smoother than individual traffic streams, rate control in response to buffer state should be easier to implement relatively.

A salient feature of the system is that the aggregation process is totally transparent to MPEG encoders and decoders as well as the network. Therefore, standard MPEG encoders and decoders can be used and network operation does not need to be modified. Furthermore, the aggregation process is a function introduced at the transmitter side so that the receivers of video streams need no changes at all.

IV. Variations of MPEG Video Aggregation System

During aggregation, some codewords of I or P frames may be discarded because of bandwidth shortage. This may causes error propagation along the corresponding sequence. According to how error propagation is dealt with, MPEG VAS's can be categorized into three classes.

For a class-A VAS, only the data of the first β codewords, which are not subject to dropping due to aggregation, are put back into the Frame Storages of the coder and decoder as the reference for interframe and interpolative coding/decoding. Since the delivery of these data is guaranteed, error propagation will not occur. However, unless β is large, less temporal redundancy can be removed by interframe and interpolative coding this way, compression becomes less efficient. A judicious choice of β is important because large β means lesser degree of aggregation, and hence potentially lesser degree of fairness among different video streams.

A class-B VAS sends feedback information to the MPEG sources as to which codewords have been chosen for delivery during aggregation, so that their respective encoders can put all of them back into the Frame Storages. Since the delivery of all forwarded data in the aggregated stream is guaranteed by the network, error propagation will not occur. Compared with class-A VAS, the feedback mechanism in class B can help increase the encoders' compression efficiency.

To further illustrate the subtlety in class-B VAS, let us consider three blocks in three successive frames. Suppose that all signals in block a (in frame 1) have been transmitted. Further suppose that block b (in frame 2) is interframe coded with motion compensation based on block a and that only the lower half (in frequency domain) of the ME errors in block b are chosen during aggregation. In response to feedback from the VAS server, the MPEG coder puts back only the lower half of that ME errors into the Frame Storage. In other words, in the Frame Storage, only the lower half signals are updated to correspond to signals in block b, while the higher half still correspond to signals in block a. As a result, when block c (in frame 3) is interframe coded, not only redundancy in the lower half signals can be removed based on block b, redundancy in the higher half signals can also be removed based on block a.

Although a class-B VAS can avoid error propagation with better compression efficiency than can a class-A VAS, real-time control of the MPEG encoders is required. It is a more cumbersome approach under certain situations. For instance, when the video sources are pre-compressed and stored in the disks for future display, a class-B VAS requires decoding and then re-coding of the video sequences during the aggregation process. On the other hand, for a VAS of class A, we may simply make sure that the pre-compressed video were coded in a way that only the first β codewords for each block are put in the Frame Storages as references.

Avoidance of error propagation as above reduces compression efficiency. A class-C VAS simply ignores, rather than avoids, error propagation. Thus, at the encoders, all data will be put into the Frame Storages (regardless of whether they have been transported). At the receiver side, all the received data will be stored at the decoders' Frame Storages. In general, for a given bandwidth, more higher-frequency components can be sent with this approach as more redundancy can be removed. However, these signals may contain propagated errors with respect to the decoders.

It is difficult to compare VAS's of class A and those of class C from the viewpoint of image quality, as this involves the comparison between degradation due to error propagation and less efficient compression, which depends to a large extent on the scene contents. Nevertheless, when the texture complexity of a video sequences is rather steady (e.g., in video-conferencing), we expect the class-C VAS's to provide better image quality. This is because as successive frames are strongly correlated, the ME error and hence degradation due to error propagation is small. By the same token, class A should perform better when successive frames are not strongly correlated (e.g., video with fast motion). Because a class-C VAS requires no modification on the "regular" MPEG encoder and decoder, we used it for our preliminary experiments described in the next section.

IV. Simulation Results

To verify that video aggregation can achieve 1) better and smoother image quality within a video stream, and 2) fairness of image quality among the video streams, some preliminary experiments have been performed. In our experiments, eight MPEG-coded color video sequences were transmitted as a bundle by a CBR channel using two approaches: 1) MPEG video aggregation, and 2) cell-level multiplexing with two-layer coding.

The resolution and frame rate of the video sequences are 320×240 and 30 frames per second, respectively (i.e., quarter size of the NTSC standard). All of them were captured from unrelated scenes in two movies, and were coded by a $N = 10, M = 3$ MPEG encoder. The traffic in terms of bits per frame of one of the sequences is shown in Fig. 4. All the traffic of the sequences is bursty in nature.

In both approaches, only the transmission of the header information, MV's and DC components is guaranteed (i.e., $\beta = 1$), while the AC codewords may be discarded when the allocated bits are not enough to accommodate all data. For aggregation, the metric used for comparing image quality of the MB's was noise energy.

For fair comparison, both aggregation and cell-level multiplexing are slotted into slice periods.[2] In each slice period, a fixed number of bits corresponding to the sum of the mean rates of the sequences are allocated.

As a simple means to reduce burstiness of the traffic to be aggregated/multiplexed, the I frames of the sessions were disaligned: the first sequence started with frame 1 (the I frame), the second sequence with frame 2, and so on. As a preliminary study, for aggregation, the effect of error propagation was simply ignored (i.e., VAS of class C was used).

[2] Buffers can be used in both aggregation and multiplexing systems to store excess data so that they can be transmitted in the next slice period (i.e., to allow bandwidth sharing across successive slice periods). In principle, this should provide better performance and would be the next set of experiments to be performed (not reported here).

A. Smoothness of Quality within a Frame

The original and the reconstructed images after multiplexing and aggregation for a frame (chosen randomly) are shown in Fig. 5. Compared with the original image (a), the post-aggregation image (c) is a little "misty", as most of the high frequency signals have been discarded. Note that, however, the quality is smooth within the whole frame. For the post-multiplexing image (b), although the left side is very well reconstructed (note the sharp edges of the table and candle), serious degradation and blocky effects can be easily discerned on the right.

If we plot the variance of the SNR of the MB's in a frame (with respect to the original frame) across the whole sequences (Fig. 6), we see that the aggregated sequence consistently has a much lower variance throughout. Thus, both subjectively and objectively, we have shown that aggregation provides much smoother quality within a frame than cell-level multiplexing does.

B. Quality Degradation due to Aggregation and Multiplexing

Subjectively, we can see from Fig. 5 that the overall quality of the post-aggregation image is superior to that of the post-multiplexing image. For an objective comparison, the SNR of all frames in one of the sequence is shown in Fig. 7. As we can see, for all frames, the post-aggregation sequence has higher SNR than the post-multiplexing sequence does. Let us define the *SNR difference* to be the difference between SNR immediately before and after aggregation/multiplexing The SNR differences of all sequences are given in Table 1. Thus, we have shown that the observation that aggregation provides better image quality than multiplexing for a given bandwidth is general to all sequences.

C. Smoothness of Quality in a Sequence

With respect to Fig. 7, image quality is steady for the sequence from cell-level multiplexing (graph (b)). Although sharp peaks occurred periodically in the one from aggregation (graph (c)), they are due to the fact that in the pre-aggregated MPEG coding, the I frames have better quality than P and B frames (graph (a)). This just shows that the aggregated frames track the quality of the pre-aggregated/multiplexed frames better. In principle,.if all the I, P, and B frames are coded to have the same quality, image qualities of both the aggregated and the multiplexed sequences will be steady, thanks to the statistical smoothing among the video sequences.

D. Fairness among the Sequences

As shown in Table 1, the standard deviation of the means of the SNR differences of the sequences is smaller in aggregation than in multiplexing. This proves that aggregation can achieve better fairness among the video sequences from the viewpoint of image quality.

V. Conclusions

This paper has investigated video aggregation, a concept that integrates compression and multiplexing of video information. It has been shown experimentally (based on the objective SNR measure and subjective observation of image quality) that video aggregation can provide better image quality than multiplexing at the cell level. Perhaps more importantly, video aggregation frees the network operator from the complicated bandwidth-allocation and tariff problems. In video aggregation, a bulk of fixed bandwidth is allocated to the group of video sessions, and it is up to the video sessions to adapt their traffic to the fixed bandwidth. Two important goals are achieved: 1) smooth image quality for the frames of each video session, and 2) fairness of image quality among the video sessions.

References

[1] M. De Prycker, *Asynchronous Transfer Mode : Solution for Broadband ISDN*, Ellis Horwood, 1993.

[2] D. Le Gall, "MPEG : A Video Compression Standard for Multimedia Applications", *Commun. of the ACM*, Vol. 34, pp.47–58, April 1991.

[3] D. Deloddere, W. Verbiest and H. Verhille, "Interactive Video On Demand", *IEEE Comm. Magazine*, May 1994.

[4] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta and J. Hill, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks", *Proc. IEEE Infocom 93*, pp. 919–926.

[5] S. S. Dixit and P. Skelly, "Video Traffic Smoothing and ATM Multiplexer Performance", *Proc. IEEE Globecom 91*, pp. 0239–0243.

[6] R. Coellco and S. Tohme, "Video Coding Mechanism to Predict Video Traffic in ATM Network", *IEEE Globecom 93*, pp. 447–450.

[7] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Commun. Magazine*, May 1994.

[8] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 3, June 1993.

[9] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks", *IEEE J. Selected Areas in Commun.*, Vol. 7, No. 5, June 1989.

[10] N. Ohta. "Packet Video : Modeling and Signal Processing", Artech House. 1994. pp. 164.

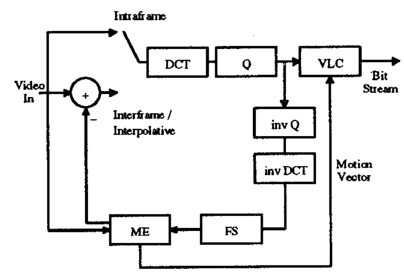
[11] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM Networks", *IEEE J. Selected Areas in commun.*, Vol. 7, No. 5, June 1989.

[12] G. K. Wallace, "The JPEG Still Picture Compression Standard", *IEEE Trans. on Consumer Electronics*, Vol. 38, No. 1, February 1992.

[13] K. R. Rao and P. Yip, *Discrete Cosine Transform : Algorithm, Advantages, and Applications*, Academic Press, Inc., 1990, pp. 170.

Table 1. Mean SNR Difference of the Sequences (in dB)

Sequence Name	*Mean SNR Difference*	
	Aggregation	Multiplexing
Death Becomes Her 1	1.26	9.50
Death Becomes Her 2	0.84	8.05
Death Becomes Her 3	0.59	8.81
Death Becomes Her 4	0.93	9.50
Far and Away 1	0.68	10.67
Far and Away 2	0.57	7.98
Far and Away 3	0.87	8.33
Far and Away 4	0.71	10.35
Standard Deviation	0.23	1.03



DCT : Discrete Cosine Transform
Q : Quantization
FS : Frame Storage
VLC : Variable Length Coding
inv DCT : inverse DCT
inv Q : inverse quantization
ME : Motion Estimation

Fig. 1. Schematic MPEG coder.

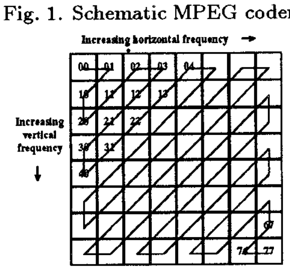


Fig. 2. Zigzag scanning order of DCT components.

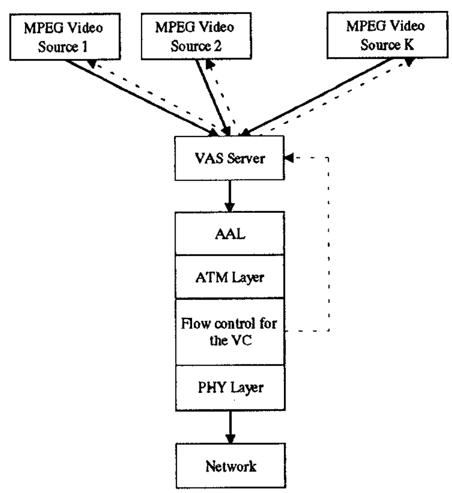


Fig. 3. Schematic diagram of a MPEG VAS. Solid arrows show the flow of data, while dotted arrows show feedback (if any).

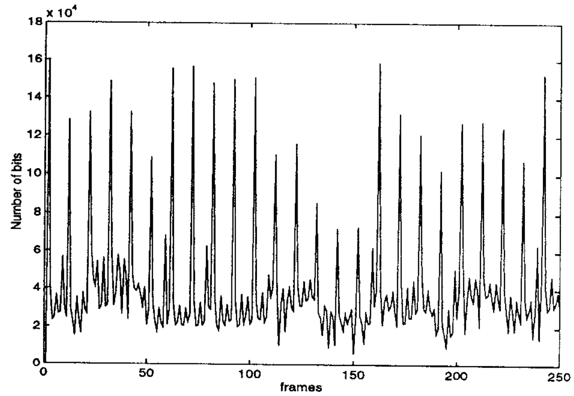


Fig. 4. Bits per frame for a sequence.

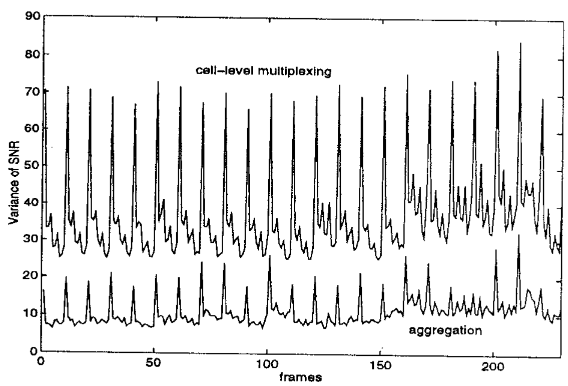


Fig. 6. Variance of SNR of the MB's in a frame within a sequence.

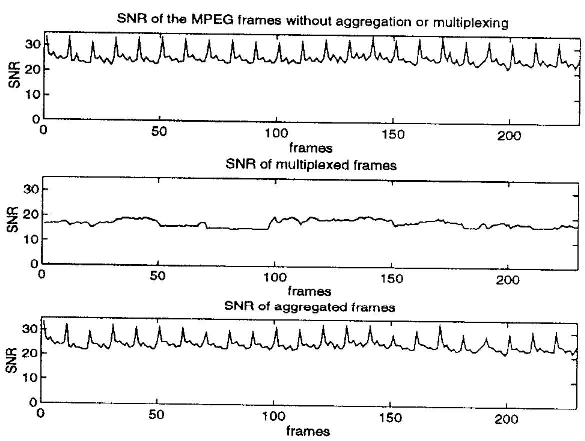


Fig.7. SNR of the frames in a sequence before and after aggregation/multiplexing.

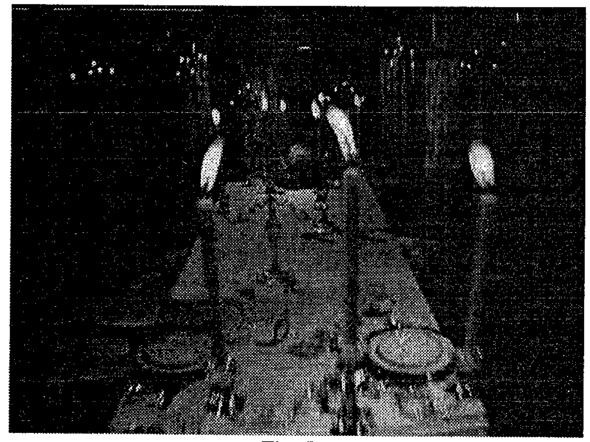


Fig. 5a.

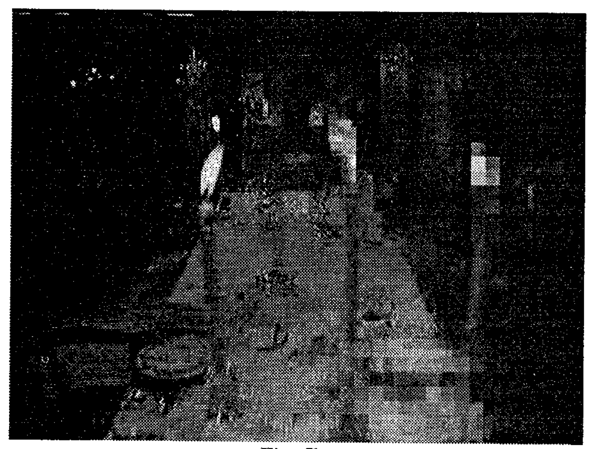


Fig. 5b.

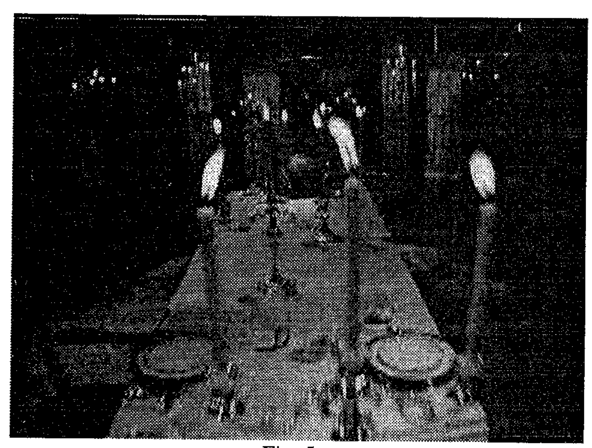


Fig. 5c.

Fig. 5. A frame in the sequence Death Becomes Her 2 (a) before MPEG coding, (b)reconstructed from the cell-level multiplexing scenario, and (c) reconstructed from the aggregation scenario.