Draft to be submitted to IEEE Transactions on Image Processing. Please send comments to Yuan-Chi Chang at yuanchi@eecs.berkeley.edu.

Delay Cognizant Video Coding: Architecture, Applications and Quality Evaluations

Yuan-Chi Chang [a], David G. Messerschmitt [a], Thom Carney [b,c] and Stanley A. Klein [b]

[a]Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley, California, USA

[b]School of Optometry
University of California at Berkeley
Berkeley, California, USA

[c]Neurometrics Institute, Berkeley, California, USA

ABSTRACT

This paper describes the architecture, applications, and quality evaluations of delay cognizant video coding (DCVC), a new type of coding algorithm for delay critical network video applications. This new coding offers many advantages over traditional algorithms, like MPEG. With the assumption of differential delay network connections, which are named delay flows, a DCVC algorithm transmits delay-critical video data through the lowest delay flow to achieve low perceptual delay. Less critical data travels through higher delay flows for a lower connection cost. The DCVC decoder employs asynchronous video reconstruction by rendering the data upon arrival. As we shall demonstrate, DCVC enables more efficient utilization of bandwidth and improves perceptual quality.

DCVC represents a new thread of layered coding targeted at multiple levels of delay services. Prior work on multi-rate and error resilient compression techniques separately address the other two quality-of-service parameters, rate and loss. We describe a networking architecture that links the three seemingly separate coding approaches into a single loosely coupled joint source channel coding framework.

The DCVC codec architecture is presented with a detailed discussion on delay segmentation and the compression of segmented video data. Subjective quality evaluations were conducted to measure the quality of asynchronously reconstructed sequences. Finally, two DCVC applications are shown to take advantage of higher delay flows for quality improvement and network capacity gain.

1 Introduction

An International Telecommunication Union (ITU) study concluded that telephony users find round-trip delays of greater than 300 ms more like a half-duplex connection than a conversation [1]. One measurement of Internet latency reported the average one-way delay can be as much as 100 ms between two nodes in the continental US [2]. Providing voice services over the Internet is clearly a great challenge. One solution taken by many switch developers is to differentiate voice and data packets and to give voice traffic preferential treatment. The high priority traffic will be charged more for the better service.

Even while the voice over IP technology is being developed, we take one step further to claim interactive video applications, such as videophone, are going to pose an even bigger challenge for the following reasons:

1. Video traffic, even after compression, requires much more bandwidth than voice. Therefore, a video call requires more network resources and will cost more.
2. Voice and video must be synchronized perceptually, which means at least a part of the video data has the same low delay requirement as voice traffic does.
3. Compressed video is very bursty (> 15: 1, peak-to-average ratio) and with the stringent delay constraint, the bursts cannot be smoothed. This implies that far more bandwidth than the long-term average rate is required.
4. The traditional video display model of frame-by-frame synchronous reconstruction requires jitter-free data alignment, which is often achieved by an artificial buffer at the receiver. Receiver buffering introduces another contribution to delay. This further tightens the end-to-end network delay budget. We can reasonably assume a tighter delay budget leads to a higher connection cost.

Although we cannot change the fact that a video connection requires more bandwidth than a voice one does, we shall reduce the magnitude of traffic bursts and relax the delay requirements of some, if not all, video data.

Our proposed solution is delay cognizant video coding (DCVC), a new type of coding algorithm for delay critical network video applications.

Abolishing the assumption that every bit in a video stream has the same delay requirement, a DCVC encoder segments video information into multiple flows with different delay requirements. This has two implications:

1. It is the lowest delay flow that determines the perceptual delay[1]. More importantly, the amount of video traffic in the lowest delay flow is smaller than it would be without the multiple flow segmentation.
2. Bursty traffic in the flows with relaxed delay requirements (other than the lowest delay flow) can be smoothed.

Both implications would result in a reduction of bandwidth usage and a lowered cost.

A DCVC decoder applies a different mode of video reconstruction from the traditional, synchronous approach. Since video packets from the transmitter are carried by differential delay flows, they arrive at the receiver at different instants. Rather than performing re-synchronization, the decoder renders the data upon arrival. Figure 1 illustrates an example of this asynchronous reconstruction, in which two video blocks acquired at the same instant in the transmitting terminal are assigned to two different delay flows in the network. The two blocks then arrive at the receiving terminal at different instants and they are displayed immediately.

Since our work on DCVC relies on network capabilities of provisioning differential delay flows, we will describe this networking framework next. With reference to this framework, Section 3 presents an overview of DCVC to outline the scope and discuss its advantages. The full codec architecture is described in Section 4. Results from subjective quality evaluation of DCVC video are discussed in Section 5. Two DCVC applications are detailed in Section 6 to demonstrate capabilities unachievable with traditional, single-stream video coding.

2 Loosely coupled joint source channel coding and the QoS flow architecture

Before getting into the technical details of DCVC, this section presents our vision of the future networking/signal processing framework and points out its link with DCVC.

[1] The perceptual delay can be quantified by the relative delay to the associated audio signal for maintaining lip synchronization. The longer the perceptual delay, the worse the interactivity of the application. Another definition is the end-to-end latency of perceiving a motion event at the transmitting end, such as nodding or gestures.

We envision the future networking environment is going to be more heterogeneous than today's network, meaning a typical connection will transverse a number of links with orders of magnitude difference in quality of service. This presents a problem to the conventional “tightly coupled” joint source channel coding (JSCC) methodology because it would be hard to make source coding adapt to all possible combinations of concatenated links. As shown in Figure 2, video coding can be tailored for a specific type of media; let it be Ethernet, asynchronous transfer mode (ATM) or code division multiple access (CDMA). Techniques such as error recovery, packetization, prioritization, and retransmission may be designed to match the characteristics of the target medium. However, when two or more links with sufficiently different quality-of-service (QoS) constitute a connection, this “tightly coupled” JSCC fails.

We have proposed in [3] a “loosely coupled” JSCC framework with differential QoS flows by adding another layer of abstraction between source and channel coding. Its reference model is shown in Figure 3. Here is a brief description on the function of each layer in this model.

- *Source coding*: Compression and encryption of basic media types such as audio, video and data. A source coder generates a set of correlated flows and assigns each flow a unique flow specification. A flow spec contains an ID, a traffic descriptor such as leaky bucket parameters, and the delay and loss requirements of the flow. This layer corresponds to an augmented presentation layer in the open system interconnect (OSI) seven layer model [5].
- *Source/transport interface*: Service and cost negotiations for each flow. The flow data structure is maintained at this interface.
- *Transport*: Networking, routing, scheduling and billing of an end-to-end flow connection. This layer corresponds to the transport and networking layers in the OSI model.
- *Transport/channel interface*: Resource negotiation and impairment allocation of the end-to-end flow spec to individual channels.
- *Channel coding*: Control and transmission of a homogeneous physical channel. This layer corresponds to the link and physical layers in the OSI model.

The reference model represents a limited decoupling of JSCC in order to tradeoff for scalability and modularity. QoS parameters shield the detailed networking implementations from upper-layer applications and yet preserve fine enough information to seize the benefits of JSCC. The communication system can still achieve the same efficiency as it would with “tightly coupled” JSCC by matching flow specs to optimized channel operations. For example, bits requiring high reliability are sent through reliable flows with forward error correction (FEC) and others are carried by less reliable flows without FEC.

Sophisticated coordination and negotiation happening at the two interfaces require a lot of message passing among different network entities. A much more efficient approach was proposed in [13] to use intelligent agents, a form of mobile codes, to speed up the process. In order to carry out the agreed QoS coordination, a subnet (channel) must also have abilities to reserve its own resource and keep track of the service received by each flow. Although today QoS guaranteed network connections are only experimental, the next generation Internet Protocol (IP) has incorporated a 28-bit flow label into its packet header structure to support future deployment [14]. The Internet Resource ReSerVation Protocol (RSVP) [15] and similar efforts for ATM are made to support QoS guaranteed flows.

Due to the limitation of space, we omitted a number of issues not directly related to this paper, such as modularity, scalability, mobility, and privacy. Interested readers are asked to consult [3][4] for details.

Even before this QoS flow architecture was proposed, video coding research has been engaged in designing multi-flow coders, in the name of layered coding, targeted at rate scalability [7]-[10] and error resiliency [10]-[12]. Although the whole framework was not specifically stated and the benefits of “loosely coupled” JSCC were not mentioned, those layered video coders have received a lot of attention due to the appreciation of the channel adaptability they enabled. Rate and error (loss/corruption) represent two of the three key QoS parameters. Delay cognizant video coding represents “the last spear of the trident” to be understood and built. Just as people started to integrate rate scalable and error resilient video coders, we believe ultimately a true QoS-adaptive video coding algorithm will be developed to incorporate all three parameters.

3 Overview of delay cognizant video coding

DCVC can be viewed as a component of the source coding layer in the aforementioned networking model. The multiple flows shown in Figure 3 can have differential attributes in delay and reliability. In this paper, we are focusing on the delay aspect.

Although the number of differential delay flows is arbitrary, there is a tradeoff between the flow structure overhead and the benefits brought by multiple flows. In the rest of the paper, we will be presenting the case that there are only two flows, the low- and high-delay flows.

The segmentation of video data into the two flows can have a significant impact on perceived quality. To minimize the visible artifacts in asynchronous reconstruction, DCVC assigns the most visually significant information to the low-delay flow and the less visually significant information to the high-delay flow. Visual significance is, at the early visual processing level, characterized by the spatio-temporal masking properties of the human visual systems (HVS), and at the cognizance level, characterized by image recognition and understanding. In our current DCVC design, we mainly make use of the former characterization of HVS masking. We consider traditional single-flow video compression such as MPEG [21] and H.261/H.263 [22] as special cases of DCVC, since they can be viewed as having two flows: one has the minimal (finite) delay and the other has the maximal (infinite) delay. The second flow simply never arrives and attributes to the quantization loss.

DCVC has a number advantages over traditional, single flow video such as MPEG and H.261. These include the increase of video traffic capacity, the reduction of perceptual delay, and the flexibility of trading off the prior two benefits. Here is the set of opportunities that MPEG does not offer.

1. DCVC reduces the amount of video traffic in the lowest delay flow without negatively affecting perceptual delay. This includes reducing the magnitude of the traffic bursts as well as lowering the average bit rate. For delay critical network video, average bit rate alone does not accurately reflect the required bandwidth. Reducing the size of the bursts is often more, if not equally, important.

2. DCVC puts a significant amount of data to higher delay flows. Traffic in these flows can be smoothed to further reduce the magnitude of the bursts.

3. For time-varying links such as wireless fading channels, higher delay flows can be buffered for the most opportune moment for transmission. For example, as channel fading strikes, the transmitter can put off the transmission of the high-delay flow and send the low-delay flow with stronger error correction codes at a lower rate. The high-delay flow can be served later when the channel condition returns to normal. But without the high-delay flow, traditional video coding is forced to take the common denominator of link rates at all fading conditions. Please note that this argument equally applies to non-bursty, constant bit rate video.

While the above three arguments have emphasized network connection cost reductions, these reductions can be redirected to further decrease the latency of the low-delay flow thereby minimizing the perceptual delay. There is clearly a tradeoff between the two goals, which we leave to negotiations at the source/transport interface in Figure 3.

4 Coding algorithm architecture

4.1 New coding problem and goals

The new video coding problem of delay cognizance no longer has a hard objective measure of delay, since visual information is generally displayed without synchronization. The focus is on perceptual delay instead, which is the amount of delay perceived by end users. With the low-delay flow carrying the most visually significant information, the perceptual delay is determined by the latency in the low-delay flow.

A DCVC algorithm attempts to optimize the following cost function while satisfying the minimum quality constraint.

$$\min_{R_1, R_2, d} C(R_1 + R_2, \frac{R_2}{R_1}, d) \quad \text{subject to} \quad Q(R_1, R_2, d) > q_0$$

R_1 and R_2 are the average bit rates of the low-delay and high-delay flows, respectively. *d* stands for the latency difference (or delay offset) between the two flows. The network cost function C increases with the total rate R_1+R_2, decreases with the ratio R_2/R_1, and decreases with the delay offset. The perceived quality function Q is increasing with R_1 and R_2, and decreasing with *d*.

The above optimization, when put in words, forms the following two objectives:

1. Minimize total compressed traffic, while maximizing the portion in the high-delay flow and minimizing the portion in the low-delay flow.
2. Maximize the allowable delay offset (the difference between the maximum allowed delay of the high-delay flow and that of the low-delay flow) that can be attained with acceptable quality.

The first objective is to reduce traffic in the low-delay flow so that minimal network resources are reserved to support its tight delay and jitter requirements. Traffic in the high delay flow has relaxed delay bounds, which gives the transport layer the most flexibility in transmission prioritizing and scheduling. Note that an implicit assumption of this objective is that the bandwidth requirement of the low-delay flow should be less than the bandwidth of the conventional, single flow video with the same quality.

The second objective is to ensure the delay relaxation is sufficient for traffic smoothing purposes. We added the constraint on acceptable visible artifacts because we expect artifacts to occur with asynchronous reconstruction. It is worth pointing out that DCVC is not another form of compression. A good compression algorithm should have removed all "invisible" artifacts subject to HVS properties and therefore, delaying the rendition of any additional information any further is going to generate visible artifacts. These artifacts become more noticeable as the delay offset increases. Unfortunately, prior HVS research has revealed little about the kinds of video information that has the most impact when delayed. It is an important issue in need of more research.

4.2 Prior work

In our earlier work, a number of different segmentation criteria were explored, but they often failed to achieve an acceptable compression ratio [16][17]. One unreported approach we tried was pixel-based segmentation by conditional replenishment, in which the basic segmentation units are pixels. In segmenting head-and-shoulder scenes, less than 5% of the total pixels are carried by the low-delay flow. The delay offset can be as great as 330 msec without incurring noticeable quality degradation[2] in our experiments when they were shown with the original video.

The compression of the 5% pixels turns out to be rather inefficient, however, because its distribution has irregular shapes and spread. The additional overhead of communicating the locations of the low-delay pixels significantly increases the amount of traffic and makes it impractical for DCVC. To reduce this overhead, the segmentation granularity is enlarged from pixels to blocks as described in this paper.

4.3 Encoding: segmentation

The DCVC encoder shown in Figure 4 is divided into two stages: segmentation and compression, each of which is framed in shaded boxes. At the segmentation stage, the encoder first divides the captured video frame into blocks of size 8 by 8. Each block is then independently assigned to the low- or high-delay flow, based on its temporal-spatial frequency variations. The flow diagram of this stage is marked and shown at the left shaded box of Figure 4.

[2] We wish to clarify that there may be no noticeable quality degradation even if one can distinguish the differences between the segmented and the original. When asked for quality judgements, observers tend to give statistically equal ratings. Therefore, the video "fidelity" differs but with indistinguishable video "quality".

There are a total of 128 test conditions, all of which must be satisfied for a block to be assigned to the high-delay flow. The 128 conditions are composed of 2 conditions each for every DCT coefficient of the tested block. Since each coefficient is independently tested, it suffices to look at just one pair of such conditions:

$$\text{Condition 1: } \left|P_{i,j,n,t} - P_{i,j,n,t-1}\right| < V_{i,j}$$

$$\text{Condition 2: } \left|P_{i,j,n,t} - P_{i,j,n,update}\right| < V_{i,j}$$

In the above expressions, $P_{I,j,n,t}$ is the (I, j)th DCT coefficient for block n at time t; $P_{I,j,n,update}$ is the (I, j)th coefficient of block n stored in a buffer for the latest update; $V_{I,j}$ is a fixed preset threshold for the (I, j)th coefficient. The 8x8 table of $\{V_{I,j}\}$ used in all the reported experiments is listed in Table 1 (for 8 bit pixels).

Table 1The 8x8 $\{V_{ij}\}$ table of DCT coefficient thresholds

30	15	15	15	15	15	30	30
15	15	15	15	15	30	30	30
15	15	15	15	30	30	30	30
15	15	15	30	30	30	30	45
15	15	15	30	30	30	45	45
15	15	30	30	30	45	45	45
15	30	30	30	45	45	45	45
30	30	45	45	45	45	45	45

The first condition is to limit the variation of a spatial frequency in two consecutive frames. The subtraction operation can be viewed as a simple 2-tap Haar filter operating in the temporal dimension. The second condition is to limit the variation relative to the latest update that is the last block assigned to the low-delay flow. The two threshold blocks in Figure 4 are marked as Condition 1 (C1) and Condition 2 (C2).

The two conditions were constructed based on our experiences in prior designs. The temporal variation of a block consists of steep changes as well as small perturbations. Steep changes typically originate from movements of objects with sharp contrast while small perturbations may come from slow variations of textures. To minimize visible artifacts, DCVC cannot ignore those steep changes and must act immediately by updating the block (region) with the low-delay flow. What DCVC can take advantage of are the small perturbations, which can be delayed in time. The first condition is to monitor steep changes of a coefficient and the second is to limit the size of perturbations. As a simple example to see their effects, a threshold of 15 is assigned to the following series of numbers.

$$\underline{40}\ 40\ 40\ 40\ 40\ 40\ 40\ 50\ \underline{60}\ 60\ \underline{76}\ 75\ \ldots$$

The numbers that trigger the thresholds are underlined.

The choice of $\{V_{I,j}\}$ directly affects the performance of DCVC. More blocks will be assigned to the high-delay flow, which is our objective, by simply increasing $\{V_{I,j}\}$. However, this would cause significant visible artifacts especially for low spatial frequencies. Through informal viewing experiments, we noticed higher spatial frequency components could tolerate a greater temporal variation without adversely affecting quality. Our observation seems to be consistent with early HVS studies on the roll-off of Contrast Sensitivity Function (CSF) at high spatial frequencies [6].

As indicated in the segmentation block diagram, two threshold units control the switch to selectively update blocks through the low-delay flow. The low-delay blocks are then inverse transformed back to the spatial domain and are put into the low-delay image plane. An example is shown in Figure 5. Typically, for head-and-shoulder scenes, the low-delay image plane consists of 10 to 20 percent of the total blocks. There is also a corresponding high-delay image plane, consisting of the blocks going to the high-delay flow. The generation of the high-delay plane is related to the compression stage, which will be described next.

4.4 Encoding: compression

The compression stage as shown at the right shaded box of Figure 4, removes spatial and temporal redundancies from the segmented video. Since redundancy removal creates data dependency, the compression of the low-delay flow cannot reference data in the high-delay flow. If the condition is violated, low-delay data cannot be decompressed ahead of high-delay data and this leads us back to the synchronous reconstruction scenario.

The compression stage complies with the dependency constraint by using two separate coding loops. Our design adopts the motion estimation (ME) with discrete cosine transform (DCT) architecture [20]-[22]. The upper (lower) ME loop in the diagram corresponds to the compression of the high- (low-) delay flow. The reconstructed low-delay image plane in the ME loop is fed back to the segmentation stage as the latest update. It is also subtracted from the original video frame to obtain the high-delay image plane. This allows quantization errors in the low-delay ME loop to be passed as a part of the input to the high-delay ME loop.

A direct compression on the image planes as they are does not turn in competitive performance because artificial block boundaries create many high frequency residues in differential coding. An effective solution, as we found, is to fill the empty regions indicated by black areas with the same blocks from the reference frame in the ME loop. Pixel values are simply copied from the reference frame to the no-value regions. Since the first frame is always intra-coded, area filling guarantees all successive frames have no empty regions.

Area filling excels over other techniques we tried for its ability to preserve the shapes of image objects and to smooth out artificial block boundaries. Blocks used in filling the empty area are not compressed for they are copied from the reference. A one-bit indicator per block is used to inform the decoder if the block belongs to the low-delay flow. At the decoder, it has the same video history as the encoder does and thus area filling can be performed without any additional overheads. For the compression of the high-delay flow, the one-bit indicator per block is not required because all blocks have high-delay components, some of which are quantization errors from the low-delay ME loop and some of which are normal video blocks.

At the outputs of both ME loops are rate control modules that monitor the output rates to ensure the compliance to QoS contract. As we are going to demonstrate in the applications of DCVC, the quantization steps of the two ME loops need not be the same and this flexibility can be exploited to further improve video quality. The rate control module in the low-delay ME loop performs adjustments of quantization steps in the conventional way. A more interesting opportunity is to make use of the module in the high-delay ME loop to control the number of high-delay blocks in a frame. With a fixed bit rate, decreasing the number of blocks to be compressed increases their coded quality. For the high-delay flow, blocks are selected based on their past history with the assumption of “delay inertia”. We presume those blocks that are less frequently updated through the low-delay flow tend to stay that way. These blocks stay on the receiver’s screen for a long time and thus deserve to have a higher quality. The rate control module keeps an age record of blocks and prioritizes the selection in the descendent order of block ages. The age of a block, incremented at each frame, is reset to zero when it is either updated as a low-delay block or selected for the high-delay flow.

The rate controlled encoding of blocks can be viewed as the video extension of progressive image transmission. A block is updated through the low-delay flow to establish a coarse representation initially. It is then replenished through the high-delay flow with a finer version. Unlike progressive image coding, video does not allow progressive coding for the whole frame.

DCVC segments the video frame to regions with different updating frequencies. For high texture, slow varying regions, the rate controlled compression proves to work well.

Finally, we applied the quantization and entropy coding tables of the videoconferencing standard H.263 [22] to our design because videoconferencing one of the target applications. The compression efficiency of both flows are listed in Table 2. Four test sequences of head-and-shoulder scenes were encoded, all of which except the Carphone sequence have still backgrounds. Compressed traffic in the low-delay flow is approximately 50 to 80 percent of the total output even though it carries only less than 20 percent of the blocks. The high portion is mainly because the compression of the high delay flow is far more effective. We expect further optimization in compression to reduce the total bit rate as well as the portion in the low-delay flow.

Table 2 Average bits per pixel for the low- and high-delay flows; raw video is captured in 12-bit resolution or YUV9 format. As a comparison, H.263 outputs are listed at the last column.

Video sequence	Low-delay flow	High-delay flow	H.263
Suzie	0.063	0.032	0.073
Salesman	0.035	0.018	0.043
Carphone	0.132	0.031	0.153
Miss America	0.028	0.024	0.034

4.5 Decoding

The DCVC decoder follows a simple set of rules to display received blocks. Compressed bit streams from both flows are tagged with frame numbers as temporal references. The decoder maintains one table for each flow, in which each entry stores the temporal reference of the received block at the coordinates. The tables are initiated to zero and blocks from earlier frames are replaced with those from later frames. By comparing $TR_{n,\ L}$, temporal reference of the *n*th block from the low-delay flow, and $TR_{n,\ H}$, temporal reference of the *n*th block from the high-delay flow, the decoder makes the following decision:

1. $TR_{n,L} > TR_{n,H}$, display the block from the low-delay flow;

2. $TR_{n,L} = TR_{n,H}$, display the sum of two blocks;

3. $TR_{n,L} < TR_{n,H}$, display the block from the high-delay flow.

These rules lead to asynchronous display of blocks. We assume when a block has its coarse representation in the low-delay flow and added details in the high-delay flow, the latency experienced by the low-delay traffic is always equal to or less than that of the high-delay traffic.

It is possible that a block from a later frame may be transmitted through the low-delay flow but it arrives at the decoder earlier than its precedents in the high-delay flow. By the above rules, this block preempts its precedents and it is rendered upon arrival[3]. The occurrence of preemption is due to significant changes of some spatial frequency components, as the segmentation is designed to detect. Movements of objects and scene changes typically cause those changes.

5 Video fidelity and quality evaluation

In spite of its promise of significant traffic capacity gain, DCVC must ensure the quality degradation to be acceptable even with a large delay offset. To verify that the performance of our coding algorithm is indeed satisfactory, we conducted both psychophysical and computational evaluations of DCVC video. Psychophysical studies rely on the participation of human subjects, who were shown video clips and were asked to judge their quality. For computational modeling, we used the traditional PSNR measure as well as a video quality metric developed by Lambrecht [30][31]. In the following, we are going to briefly describe the experiments on video fidelity and focus more on quality tests as well as computational modeling.

5.1 Video fidelity

In these experiments, we are interested in knowing if sequences with nonzero delay offset between the two flows can be visually discriminated from the original, jitter free video rendering. The video sequences used in the experiments were the luminance components of standard H.263 test clips: Suzie, Salesman, and Mother-daughter (see Appendix A: Video evaluation methods for details). Both compressed and uncompressed sequences were used. A standard self-paced psychophysical method of adjustment was used with 3 delay offsets (0, 1, and 2 frames) for the high delay flow. Each run consisted of 100-150 trials with correct response feedback provided after each trial.

We found judgements on video fidelity to be unanimous: for both uncompressed and compressed video sequences, asynchronous rendering does not preserve video fidelity. Even a delay offset of one frame can be detected. This result is not surprising since aggressive

[3] It is conceivable that further traffic capacity gain may be obtained by instructing the networking layer to stop forwarding those blocks in the high-delay flow, which are now obsolete. The annihilation of the blocks is, however, not possible in our reported design using differential coding. If they are dropped in the network, the motion compensation loop of the high-delay flow at the decoder will lose synchronization with the loop at the encoder. The annihilation can be made possible by removing the dependency and using non-differential coding for compression. Special packetization and application-aware routers must be deployed to take advantage of block preemption.

segmentation ought to create discriminable visual differences. While conducting the experiments, we also observed a learning effect. After a subject watched the same sequence 20 to 30 times, they learned to focus on specific details in the sequence for making the discrimination and ignored the rest of the image. A conferencing participant is not likely to watch the video more than once so this type learning is unlikely to occur in practical application of DCVC. Therefore, we are less concerned about fidelity than perceived quality.

Another interesting observation was that while delays could be detected, the video quality of compressed sequences did not necessarily degrade. In fact, for some compressed sequences and some observers, the quality appeared to improve with relatively long delays. This effect appears to be related to a reduction of mosquito noise when delay is introduced. This observation prompted us to examine video quality in more detail.

5.2 Video quality

Unlike the fidelity experiments, this set of experiments focused on evaluating the quality of compressed video only. In particular, we are interested in the compression-introduced masking effect on the nonzero delay sequences. From the fidelity experiments, we know that nonzero delay offset and its accompanied asynchronous reconstruction introduce visual differences in uncompressed sequences. It is also known that lossy compression generates quantization noise. Furthermore, we observed that the noise contributed by compression seems to be stronger and dominates the perception of the overall video quality. Our goal in these experiments is to characterize the effect of delay on compressed video quality. The procedure of the experiments and the preparation of test sequences are described in Appendix A.

Video Quality Ranking: The results from ranking the four simultaneously presented stimuli, steps 1 and 3 from Appendix A, are summarized in Table 3. The numbers in the table represent the frequencies of stimuli being ranked as the best, 2nd best, 3rd best and worst video quality when the four stimulus conditions are compared to each other. The table notations for the four stimulus conditions; high compression with no delay, high compression with twelve frame delay, low compression with no delay and low compression with twelve frame delay, are H_0, H_{12}, L_0 and L_{12}, respectively. The table frequencies represent the aggregated rankings given by the eleven subjects for the seven video sequences. Because every subject ranked each sequence 4 times in an experiment, the total number of data points per condition is 308.

Table 3 Aggregated ranking results from 11 subjects and their choices on 7 test sequences

Votes	L_0	L_{12}	H_0	H_{12}

Votes	L_0	L_{12}	H_0	H_{12}
Best	125	164	4	15
2nd Best	134	106	25	43
3rd Best	39	32	106	131
Worst	10	6	173	119

Larger numbers in this table are distributed in the upper-left and lower-right quadrants. As expected, the difference in compression level between first two columns and the second two columns had a significant impact on video quality. The impact of delaying part of the video stream on video quality is more subtle but also significant none the less. The difference between the high compression conditions, with and without delay (H_0 & H_{12}) was significant ($X^2 = 23.8$; $p < 0.01$). Similarly, the difference between the low compression conditions, with and without delay (L_0 & L_{12}) was also significant ($X^2 = 10.2$; $p < 0.05$).

Consider the high compression conditions, the response distribution in the H_{12} condition column is shifted up relative to the distribution in the H_0 column which indicates the delayed video was favored over synchronous video. Similarly, for the video sequences with less compression, the response distribution in the L_{12} column is shifted up relative to the distribution in the L_0 column which indicates the delayed video had higher quality than the normal or non-delayed video. This result is most surprising, delaying part of the video stream improved video quality for compressed H.263 video sequences. DCVC can improve network performance and improve video quality at the same time, a finding that has important implications for future low bandwidth video coding.

Video Quality Ratings: The video quality ranking results were unexpected. If delay improves video quality in side by side comparisons, would it still be observable when sequential video quality assessments are made? Two sets of quality ratings using the same stimuli were gathered about 30 minutes apart for each subject. Test-retest analysis of the two data sets indicates they can be safely combined into one data set.[4][28][29]

Since video content might have impact on video quality for DCVC sequences, we evaluated the effect of DCVC separately for each sequence. A three-way repeated measures analysis of variance (treatments-by-treatments-by-subjects design) was performed on each of the seven video sequences. The results for each video sequence are shown in Table 4. The first three columns contain the F-ratios for the main effect factors: compression, delay, and subject respectively. Subject is incorporated in the analysis because observers applied different ranges of values.

[4] We found 88.6% of the 308 repeated measure pairs were not significantly different at $p < 0.05$

Factors that had a significant effect on video quality are indicated by an asterisk. The last four columns contain the mean video quality rating for the four conditions. For all video sequences, the compression level has a significant effect on video quality. However, the delay factor was significant ($p<0.05$) for two sequences, the Carphone and Salesman sequences. For the Carphone sequence, eleven-frame DCVC delay improved image quality. However, for the Salesman sequence, the same delay degraded the image quality. For the rest of the sequences, delay had no significant effect. In general, long delay offset has limited influence on perceived video quality, either positively or negatively. Lastly, the subject factor is always significant as we expected since different observers applied different rating ranges.

Table 4 Results from three-way effects analysis of variance on the ratings aggregated across 11 subjects

Sequence	F_{rate}	F_{delay}	$F_{subject}$	L_0	L_{12}	H_0	H_{12}
Carphone	205.60*	4.38*	8.25*	4.60	4.93	2.48	2.77
Foreman	309.52*	2.61	10.04*	5.55	5.11	2.92	2.91
Salesman	76.99*	3.96*	4.17*	5.00	4.52	3.48	3.34
Suzie	249.17*	0.36	15.19*	5.10	5.00	2.85	2.78
Mother	130.58*	0.71	13.68*	4.40	4.31	2.41	2.76
Claire	190.86*	0.24	12.51*	5.15	5.23	3.24	3.30
Miss Am	87.32*	1.90	11.27*	4.33	4.61	3.07	3.18

(* = $p< 0.05$)

Results from quality ranking evaluations using simultaneous presentation of all four conditions indicated that delayed video looked the same or better than traditional, synchronous video. When observers were asked to make quality ratings for the same sequences presented one at a time, the improvement with delay disappeared for all but one video sequence. Video quality was generally not effected by the large twelve frame delay. The lack of having a direct comparison stimulus and having to rely on memory probably accounts for the improvement with delay effect disappearing in the sequential testing conditions.

How can delay improve video quality, even by a small amount? Figure 6 schematizes an example where delay should improve video quality by reducing dynamic noise. This condition occurs when an original, uncompressed block (8x8 pixels) is varying slowly in time. Under high compression, the compressed block contains quantization noise. Upon rendering the video sequence, the block closely follows the luminance variation of the original block. However, the quantization noise changes from frame to frame as shown in the second row of Figure 6. This noise is often referred to as mosquito noise and is very annoying. For delayed video, however, the encoder sends the second to the fourth block to the high delay flow, which will arrive at the receiver after 400 msec. In the meaning time, the decoder simply keeps showing the first block received as shown in the third row of Figure 6. The quantization noise seen by our subjects is thus static. The static noise is preferred to the dynamic noise. The static noise might even be attributed to the original image but dynamic noise is clearly not a part of the original scene. Presumably, this discovery can be well applied to MPEG and other compression algorithms because skipping the blocks that cause dynamic noise improves quality and reduces bit rate at the same time. We will be exploring this idea in future works.

5.3 Computational modeling methods

We employed the popular PSNR measure as well as a computational vision model developed by Lambrecht [30][31] to quantify video quality. PSNR is the commonly used metric for its simplicity and universal mean squared error formulation. The Lambrecht model, named Moving Pictures Quality Metric (MPQM), was developed for evaluating perceptual quality of video sequences based on a spatio-temporal model of human vision. It includes a multi-scaled arrays of Gabor shaped spatial filters at several orientations, intra-channel masking and a Minkowski summation stage. The model also includes an extension to the time domain by adding sustained and transient temporal filters to evaluate video sequences. Video inputs to both metrics were adjusted to reflect the CRT luminance nonlinearity (gamma function) to approximate what human subjects saw on the screen.

Table 4 Peak Signal-to-Noise Ratio per frame of the Mother-daughter sequence

PSNR(dB)	L_0	L_{12}	H_0	H_{12}
Average	32.54	32.45	31.72	31.66
Minimum	32.17	32.08	31.29	31.3
Maximum	33.03	32.93	32.2	32.19

Table 4 listed the average, minimum and maximum of the per-frame PSNR of the four stimulus conditions in the Mother-daughter sequence. Similar results apply to other sequences. The PSNR measure predicted nonzero delay offset sequences had a lower quality by a relatively small amount of difference. The difference is much bigger when PSNR is calculated based on blocks, which probably reflects human perceived quality better. The per-block PSNR difference can be as much as 1.46 dB as shown Table 5, which again predicts the quality degrades in asynchronous reconstruction. As a reference, Table 6 lists the per-block PSNR differences of H_0 and L_0, which have different compression levels.

Table 5 Maximum PSNR difference between H_0 and H_{12} calculated per-block.

Sequence	Mother-daughter	Miss America	Suzie	Foreman
DPSNR(dB)	1.46	1.16	0.7	1.04

Table 6 Maximum PSNR difference between L_0 and H_0 calculated per-block

Sequence	Mother-daughter	Miss America	Suzie	Foreman
DPSNR(dB)	2.12	2.62	2.97	2.96

The outputs of MPQM are quantified in just-noticeable-distortion (JND). Table 7 listed the average, minimum and maximum of the JND values in MPQM units for the Mother-daughter sequence. Like the PSNR metric, MPQM also did not predict that delayed video looked better. The results showed that the effect of delay contributed approximately 30 percent of degradation relative to the change of compression levels.

Table 7 MPQM outputs of the Mother-daughter sequence; the higher the noise, the lower the quality.

JND	L_0	L_{12}	H_0	H_{12}
Average	0.207	0.216	0.236	0.245
Minimum	0.187	0.194	0.210	0.218
Maximum	0.230	0.240	0.262	0.269

We were hoping to replace time-consuming psychophysical experiments with computational metrics, the above results demonstrated the time has yet to come. Neither the commonly used PSNR nor the HVS based MPQM adequately captured the effect of differential delay. Further enhancement of the HVS modeling methods should be addressed.

6 DCVC applications

The development of DCVC enabled a number of new video applications that were not achievable in the traditional, single flow, synchronously reconstruction configuration. In this section, we are going to demonstrate two of them; the first application is to improve network video quality at no additional cost; the second one is to increase the transport traffic capacity.

6.1 Improving network video quality

Our first DCVC application, first described in [27], was motivated by the observation that for variable bit rate (VBR) video streams, typically significantly more bandwidth than the long-term average rate is reserved to guarantee a low delay transport delivery. In the networking research community, this problem has long been recognized and numerous works have been done to estimate the effective bandwidth of a VBR video connection[23]-[26]. Effective bandwidth, or sometimes referred as equivalent capacity, of a stochastic source characterizes the bandwidth needed to guarantee a small, nonzero packet loss probability (typically 10^{-5} or lower) in a switching node. By allowing a small loss probability, more video connections can be statistically multiplexed to the same link thereby improving traffic efficiency. Due to the bursty nature of video, this traffic capacity gain can be significant. Effective bandwidth is a function of this loss probability, the size of the switching buffer, and the stochastic property of the traffic source. Always less than the peak rate of a VBR source, effective bandwidth is still greater than or equal to the average rate.

In traditional video coding, the difference (residual bandwidth) between the effective bandwidth and the average rate is taken for granted, as it is required for statistical multiplexing. Few if any prior research proposed to make use of the residual bandwidth. However, in future pay-for-service networks, end users will be charged based on the reserved bandwidth, which includes residual bandwidth. Consequently, the additional bandwidth is reserved and billed, but never used.

DCVC created an opportunity to make use of residual bandwidth. The delay critical information is carried by the low-delay flow to establish an initial image while the high-delay flow improves the video quality in a progressive manner. Since the high-delay flow has a relaxed delay requirement, it can be fit into the residual bandwidth with proper rate control. The effective bandwidth of a DCVC connection is computed and reserved solely based on traffic statistics of the low-delay flow. The use of residual bandwidth does not affect the formulation or the outcome of statistical multiplexing analysis. A simple priority assignment can accommodate DCVC traffic. Whenever the switching capacity is available, packets carrying the low-delay flow always have higher transmission priority over those carrying the high-delay flow. The switching node can thus be viewed as a prioritized two-class, single server queue.

The video quality improvement comes at no additional cost to end users, since the residual bandwidth has been charged as a part of the reservation. The free improvement would not be possible with traditional, single flow video because the residual bandwidth is unpredictable. It varies in time and is closely associated with other traffic connections multiplexed at the switching node. A video encoder typically has no access to traffic patterns of other coexisting connections and thus does not know the exact amount of additional capacity. Rather than attempting to predict the instantaneous value, a DCVC encoder allows the high-delay traffic to be shaped to fit in the time varying residual capacity.

To demonstrate the quality improvement with DCVC, we encoded a 15-second video sequence and simulated its transmission through a network switch. The sequence contains 450 frames and is a concatenation of three short clips with 150 frames each. Although the average bit rate of the low-delay flow is 30 Kbps, its peak rate is almost 17 times of that due to intra-coding at scene changes. Rate control was applied to the high-delay flow to reduce the number of blocks encoded at each frame and to increase the quality of those that are encoded. We consider a network switch with a capacity of 1.5 Mbps. The maximum queuing delay of the low-delay flow is set to be 400 ms, which is equivalent to a buffer size of 600 Kbits. The packet loss probability must be 10^{-6} or lower.

The low-delay flow of the compressed video sequence is approximated by the two-state Markov model described in Appendix B. Parameters used in computing its effective bandwidth are: μ_1 = 26.5 Kbps; μ_2 = 512 Kbps; p_{12} = 1/149; p_{21} = 1; $\delta = 2.3*10^{-5}$. The effective bandwidth of the low-delay flow is 161 Kbps, less than ten of which can be admitted to the network switch simultaneously. With random starting points through the duration of video, we first simulated ten of such sequences with the low-delay flow only and observed no violations on the given loss probability. We then added the high-delay flow traffic to our simulation to observe its maximum queuing delay. Although the high-delay flow has a lower transmission priority, the several hundred simulations we performed showed the maximum waiting time in this lower priority queue never exceeds 90 msec.

We are interested in comparing the perceived video quality of a sequence with the low-delay flow only and that of a sequence with both flows. Despite that the actual delay experienced by the high-delay traffic may be time-varying, we consider the worst case in which all the packets in the flow lag their counterparts in the low-delay flow for 99 msec. We found the informal subjective evaluation by graduate students to favor the two-flow DCVC-coded video. We also computed the peak signal-to-noise ratio (PSNR) of both conditions when compared with the original. As shown in Figure 7, the PSNR improvement which is always positive, sometimes even exceeds 2dB.

6.2 Increasing network capacity

Our second application demonstrates that DCVC delivers the same quality at a lower effective bandwidth thereby increasing the traffic capacity of the network. Recall that in the first application, as much as 2dB increase in quality, measured in PSNR, can be achieved through the addition of the delay-tolerant high-delay flow. Our approach to increase capacity is to convert the 2dB quality gain into bit savings. As a traditional, single flow video encoder compresses the video at the quantization level N, the two-flow DCVC encoder assigns a level greater than N to the low-delay flow and a level less than N to the high-delay flow. A high quantization level degrades quality but reduces the bit rate. A DCVC encoder adjusts both levels to deliver the same quality as the single-flow case. The effective bandwidth of DCVC video can be shown in the following example to be 30% or less than the effective bandwidth of traditional video. Therefore, for every two traditional video connections carried in the network, they can be more efficiently replaced by three DCVC connections. This is equivalent to an increase in traffic capacity by 50%.

We used a H.263 coder and the DCVC coder to encode the 15-second Salesman sequence. The quantization level of the H.263 coder was set to 8. The compression level of the low-delay flow of DCVC was set to 10 while that of the high-delay flow was set to 5. Rate control was activated for the high-delay flow to limit that a maximal 10% of the frame area is encoded. Rate control is necessary to adjust the bit rate of the high-delay flow to fit in the residual bandwidth of the low-delay flow. The PSNR measured quality of the H.263 stream and the quality of DCVC video with a 10-frame delay offset is plotted in Figure 8. DCVC video has a higher PSNR most of the time, except for the first 20 frames, where the low-delay flow establishes a coarse representation and waits for the high-delay flow to gradually improve the quality.

We again applied the two-state Markovian traffic model described in Appendix B. The effective bandwidth of the H.263 stream is estimated to be 158 Kbps. The effective bandwidth of the DCVC low-delay flow is 110 Kbps, which has sufficient residual bandwidth to carry the high-delay flow. As shown from these numbers, DCVC requires 30% less bandwidth to deliver the same quality.

7 Conclusions

We presented a new, delay cognizant perspective for video coding and demonstrated a DCVC design that delivers good subjective quality even with long delay offsets. Motivated by the mismatch of service provision between modern packet networks and traditional, synchronously rendered video, DCVC, as we pointed out in the paper, is an integrated component of QoS adaptive schemes traced back to multi-rate coding. Despite significant progress made since it was first proposed, there are still a number of open issues highlighted in this article. While these minor, yet crucial, improvements remain to be refined further, we wish to point out two more grand challenges: one to the video coding research and the other to the networking research. For video coding, the integration of *rate scalability*, *error resilience*, and *delay cognizance* into a single coding algorithm will enable the full QoS abstraction of video flows. As prior work, including ours, have been focusing on one or two aspects, a direct extension to all three may not be as straightforward and further studies are required. For networking, the exploitation of the QoS adaptive flows proves to be another interesting topic. A switch node should optimize its decision on when to delay transmissions, what flows to suspend and which packets to drop. Again, published works have been focusing on only one or two aspects of the problem. We hope to see an integrated solution to be proposed.

Acknowledgements

The authors would like to thank Prof. Murat Kunt and Dr. Christian J. van den Branden Lambrecht of the Signal Processing Lab at Swiss Federal Institute of Technology in Lausanne, Switzerland, for providing the Moving Pictures Quality Metric (MPQM) system. This research was supported by the University of California MICRO program, grants from Harris, Rockwell, LG Electronics, National, Plessey, Philips, and an IBM Fellowship. The research was also supported by grant from the Air Force Office of Scientific Research (F49620-95) to the Neurometrics Institute.

References

[1] ITU Recommendation G.114, "One-way Transmission Time," *International Telecommunication Union*, Feb. 1996.

[2] T. J. Kostas, et al., "Real-time voice over packet-switched networks," *IEEE Network Magazine*, pp. 18-27, Jan. 1998.

[3] P. Haskell, D. G. Messerschmitt and L. C. Yun, "Architecture principles for multimedia networks", *Wireless Communications: Signal Processing Perspectives*, H. V. Poor and G. W. Wornell, Ed., Prentice Hall, 1998.

[4] L. C. Yun and D. G. Messerschmitt, "Digital video in a fading interference wireless environment," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Proceesing*, pp.1069-1072, Atlanta, GA, 1996.

[5] M. Schwartz, "Telecommunication networks: protocols, modeling and analysis," published by *Addison-Wesley*, 1987.

[6] B. Wandell, "Foundations of vision," published by *Sinauer Associates*, 1995.

[7] J. Y. Tham, S. Ranganath, and A. A. Kassim, "Highly scalable wavelet-based video codec for very low bit-rate environment," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 1, pp.12-20, Jan. 1998.

[8] D. Taubman and A. Zakhor, "Multirate 3D subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572-88, Sep. 1994.

[9] J. Ohm, "Advanced packet-video coding based on layered VQ and SBC techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 3, pp. 208-21, June 1993.

[10] K. Ramchandran, A. Ortega, K. Uz and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 1, pp. 6-22, Jan. 1993.

[11] W. S. Lee, M. R. Pickering, M. R. Frater, and J. F. Arnold, "Error resilience in video and multiplexing layers for very low bit-rate video coding systems," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp.1764-74, Dec. 1997.

[12] E. Steinbach, N. Farber, and B. Girod, "Standard compatible extension of H. 263 for robust video transmission in mobile environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 6, pp. 872-81, Dec. 1997.

[13] Wei-Yi Li, “Agent-augmented network signaling for call setup,” *Ph.D. dissertation*, University of California at Berkeley, 1998.

[14] S. Bradner and A. Mankin, “The recommendation for IP next generation protocol,” *IETF RFC 1752*, 1995.

[15] R. Braden et al., “Resource Reservation Protocol (RSVP) – version 1, functional specification,” *IETF RFC 2205*, 1997.

[16] J. Reason, L. C. Yun, A. Lao, D. G. Messerschmitt, “Asynchronous video: coordinated video coding and transport for heterogeneous networks with wireless access,” *Mobile Wireless Information Systems*, Kluwer Academic Press, 1995.

[17] Y. C. Chang and D. G. Messerschmitt, “Delay cognizant video coding,” *Proceedings of International Conference on Networking and Multimedia*, Kaohsiung, Taiwan, pp. 110-117, 1996.

[18] Y. C. Chang and D. G. Messerschmitt, “Segmentation and compression of video for delay-flow multimedia networks,” *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, 1998.

[19] Y. C. Chang, T. Carney, S. A. Klein, D. G. Messerschmitt, and A. Zakhor, “Effects of temporal jitter on video quality: assessment using psychophysical methods,” *Proceedings of the SPIE – Human Vision and Image Processing*, San Jose, CA, 1998.

[20] R. J. Clarke, “Digital compression of still images and video,” published by *Academic Press*, 1995.

[21] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg and D. J. LeGall, *MPEG video compression standard*, Chapman & Hall, 1997.

[22] “Video coding for low-bit rate communications: draft recommendation ITU-T H.263,” International Telecommunications Union - Telecommunication Standardization Sector, May 1996.

[23] R. Guerin, H. Almadi, and M. Naghshineh, “Equivalent bandwidth and its application to bandwidth allocation in high-speed networks,” *IEEE Journal on Selected Areas in Communication*, vol. 9, no. 7, pp. 968-981, 1991.

[24] A. I. Elwalid and D. Mitra, “Effective bandwidth of general Markovian traffic sources and admission control in high-speed networks,” *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 329-343, 1993.

[25] D. N. C. Tse, R. G. Gallager, and J. N. Tsitsiklis, “Statistical multiplexing of multiple time-scale Markov streams,” *IEEE Journal on Selected Areas in Communication*, vol. 13, no. 6, pp. 1028-1038, 1995.

[26] G. Kesidis, J. Walrand, and C. S. Chang, “Effective bandwidth for multiclass Markov fluids and other ATM sources,” *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 424-28, Aug. 1993.

[27] Y. C. Chang and D. G. Messerschmitt, “Improving network video quality with delay cognizant video coding,” *Proceedings of IEEE International Conference On Image Processing*, Chicago, IL, 1998.

[28] T. W. Anderson and J. D. Finn, “The new statistical analysis of data,” published by *Springer-Verlag*, 1996.

[29] B. E. Wampold and C. J. Drew, “Theory and application of statistics,” published by *McGraw-Hill*, 1990.

[30] C. Lambrecht and O. Verscheure, “Perceptual quality measure using a spatio-temporal model of the human visual system,” *Proceedings of the SPIE* vol. 2668, San Jose, CA, pp.450-61, 1996.

[31] C. Lambrecht, “A working spatio-temporal model of the human visual system for image restoration and quality assessment applications,” *Proceedings of IEEE Int. Conf. On Acoustics, Speech, and Signal Processing*, Atlanta, GA, pp. 2291-4, 1996.

Appendix A: Video evaluation methods

Eleven paid volunteer subjects from the UC Berkeley campus participated in the experiments in January, 1998.

The seven raw video sequences used in the experiments are standard H.263 test clips: Carphone, Claire, Foreman, Miss America, Mother-daughter, Salesman, and Suzie. The test sequences are available from ITU and its members. They are stored in the 4:2:0 QCIF format (176 by 144 pixels). For both the fidelity and quality experiments, only the luminance component of the video was used. Each sequence was 2.5 seconds long (75 frames) and was presented on a Sony Trinitron monitor at 60 Hz (two scans per frame). MATLAB with the PC-MatVis (www.neurometrics.com) psychophysical testing and stimulus presentation extension were used to present the stimuli and gather the rating data. Among the encoded sequences, the number of low-delay blocks was between 10 to 20 percent of the total. The actual percentage is content dependent. For nonzero delay offset sequences, we applied the same amount of delay uniformly, in the units of frame display time (1/30th of a second), to the video data in the high-delay flow.

The test sequences were generated with two independent variables, delay and compression. Each variable had two levels for a total of four stimulus conditions per sequence. We investigated not only the effects of delay offset but also the effects of compression-introduced masking.

- *Compression level*: The first frame (the only I frame) of all four stimulus conditions was compressed at the same quantization level and thus contained identical information. The amount of compression-introduced noise in subsequent frames is controlled by the quantization level (QL). All 64 DCT coefficients of inter-coded blocks are quantized with the same level. Increasing the quantization level decreases the video quality and vice versa. In stimulus conditions 1 and 2, QL was set to 10 for all seven sequences. In stimulus conditions 3 and 4, QL was set to 12 to compress Salesman, Mother-daughter, and Miss America while the other four were compressed with QL equal to 13. Depending on the video content, a decrease of QL from level 12 to 10 increases the compressed bit rate by 20 to 50 percent.
- *Delay level*: Synchronous, zero-delay-offset video reconstruction was applied to conditions 1 and 3. A delay offset of 12 frames (~400 milliseconds) between the low- and high-delay flows was applied to stimulus conditions 2 and 4. Nonzero delay offsets lead to asynchronous video reconstruction.

The procedure of evaluating video sequence quality involves the following three steps.

1. *Simultaneous Presentation Quality Ranking*: All four stimulus conditions (video clips) were presented simultaneously, two across and two down on the screen. Stimulus locations were chosen randomly. The 2.5-sec long presentation was repeated ten times (additional viewing time was available as desired by the subject). The subjects were asked to rank order the four stimuli using their own subjective criteria for quality.
2. *Successive Presentation Quality Rating*: Each of the four stimulus conditions was presented individually in random order for a total of 20 trials, 5 for each condition. Each stimulus presentation lasted 5 seconds (two repeats). After each stimulus presentation, the subject was asked to rate the image quality on a scale of 0 to 9. Subjects were not told that only the four stimulus conditions seen earlier were being presented again. They were told that the four stimuli that appeared in step 1 bracketed the range of quality levels to be presented in this step of the experiment.
3. *Repeat*: Finally, step 1 above was repeated using the same stimulus conditions. Subjects were not informed that the stimulus conditions in steps 1 and 3 were in the same screen locations.

To evaluate consistency of the subject responses, the three steps above were performed for all seven sequences and then repeated. It took each subject about an hour to finish the experiment.

The most often received comment from our subjects was the difficulty in rating video quality in step 2. With highly compressed sequences, different patterns of noise appeared in different parts of the image and were varying over time. In preliminary studies, when step 2 was performed alone subject rating criteria for video quality appeared to shift over time generating “inconsistent” results. We found that step 1 helped in reducing the inconsistency by presenting four stimuli simultaneously. The longer viewing time gave subjects an opportunity to study the stimuli and establish stable criteria.

Appendix B: Effective bandwidth of a two-state Markov modulated fluid model

In the following, one effective bandwidth formulation is briefly described. Significant amount of research on effective bandwidth in recent years leads to a number of different formulations and proposals [23]-[26]. While these works differ in the stochastic models for traffic streams, they are essentially based on *large deviation* estimates of the loss probability in the asymptotic regime of large buffers. As the buffer size increases, the loss probability approaches zero at an exponential rate. As one might have expected, for a fixed buffer size, the effective bandwidth of a source approaches its peak rate when this probability decreases to zero.

Consider a Markov-modulated model for a video stream with different bit rates in different states. Let the required loss probability be expressed as $e^{-\delta B}$, where B is the buffer size. It is shown in [25] that the effective bandwidth of the model can be expressed as $\Lambda(\delta)/\delta$, where $\Lambda(\delta)$ is the log spectral radius function of the matrix $[P_{ij}e^{\delta\mu_i}]$. P_{ij} is the transition probability from state i to state j and μ_i is the bit rate at state i.

The compressed video streams used in our experiments can be approximated by a two-state Markov-modulated model. The above stochastic matrix can be shown to be:

$$\left[P_{ij}e^{\delta\mu_i}\right]=\begin{bmatrix}(1-p_{12})e^{\delta\mu_1} & p_{12}e^{\delta\mu_1}\\ p_{21}e^{\delta\mu_2} & (1-p_{21})e^{\delta\mu_2}\end{bmatrix}$$

Its log spectral radius function is the logarithm of the largest positive eigenvalue, which has a simple closed form solution for this 2x2 matrix.

$$\Lambda(\delta)=\log\frac{b(\delta)+\sqrt{b^2(\delta)-4a(\delta)}}{2}$$

$$a(\delta)=(1-p_{12}-p_{21})e^{\delta(\mu_1+\mu_2)};\ b(\delta)=(1-p_{12})e^{\delta\mu_1}+(1-p_{21})e^{\delta\mu_2}$$

One set of values for these parameters are given in Section 6.

Figures

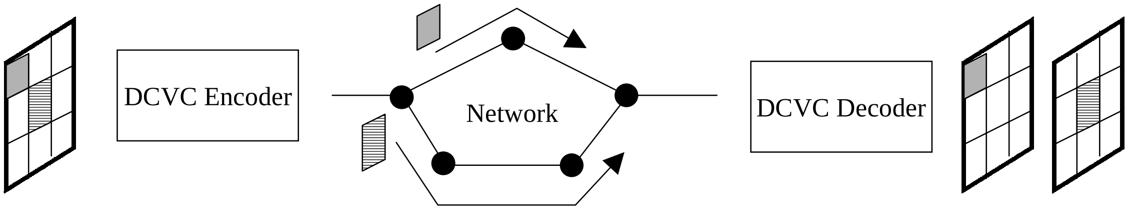

Figure 1 An illustration of differential delay flows and asynchronous reconstruction; the top path is shorter and thus the block arrives earlier than the other.

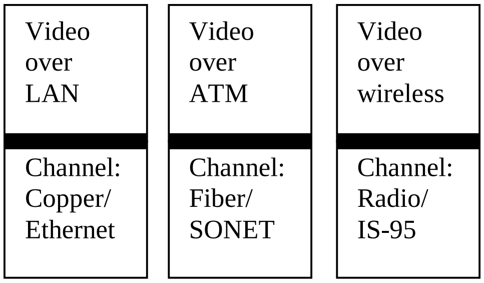

Figure 2 An example of tightly coupled joint source channel coding. Published research on video has designs tailored for specific media such as video over LAN, ATM, or wireless.

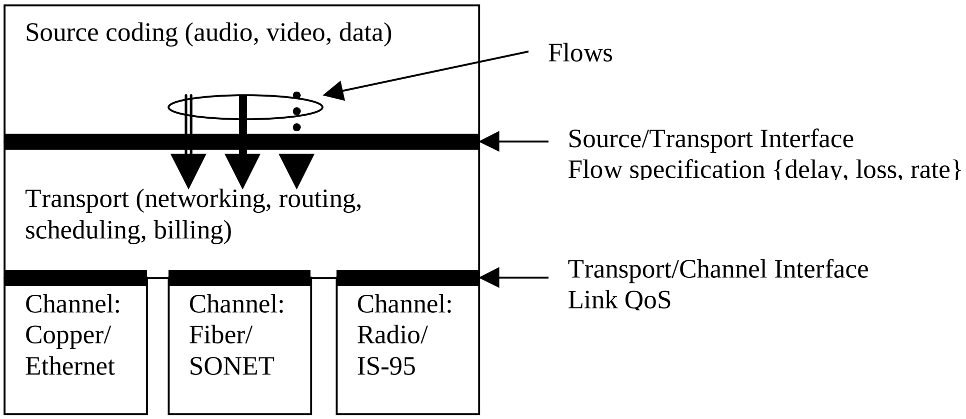
Source coding (audio, video, data)
Flows
Source/Transport Interface
Flow specification {delay, loss, rate}
Transport (networking, routing, scheduling, billing)
Transport/Channel Interface
Link QoS
Channel: Copper/ Ethernet
Channel: Fiber/ SONET
Channel: Radio/ IS-95

Figure 3 The three-layer architecture reference model for loosely couple joint source channel coding.

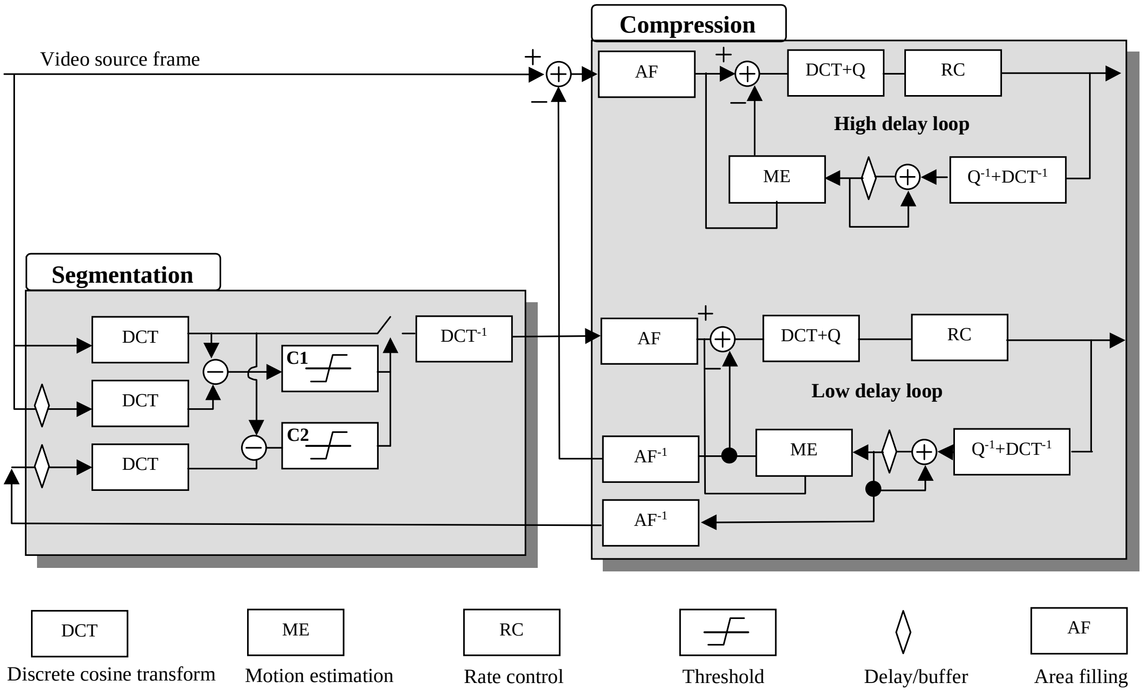

Figure 4 DCVC encoder block diagram; the two stages are framed in shaded boxes; -1 appeared in a function block represents inverse operations.

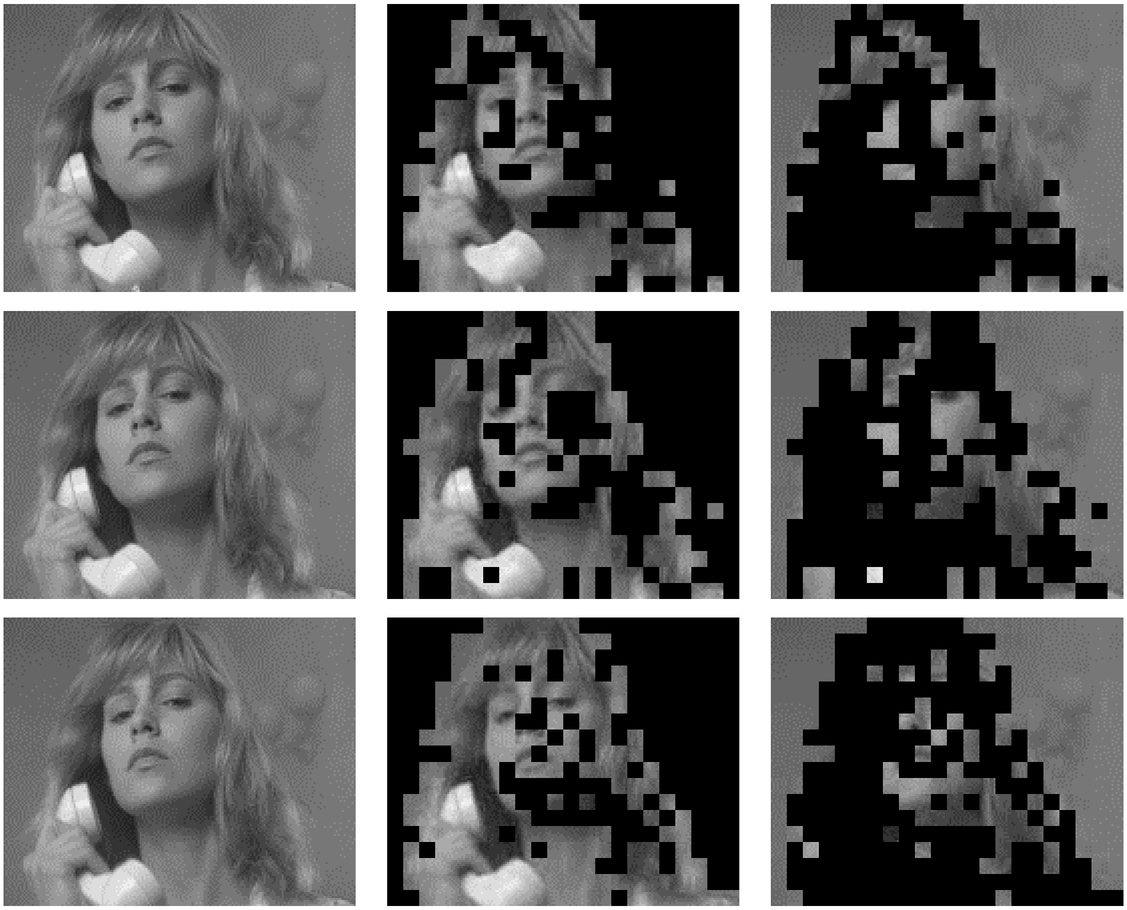

Figure 5 The original (left), the low-delay (center) , and the high-delay (right) image planes of three consecutive frames of the Suzie sequence. The empty regions are shown in black. Note that for clarity in comparison, area filling was not applied to these images.

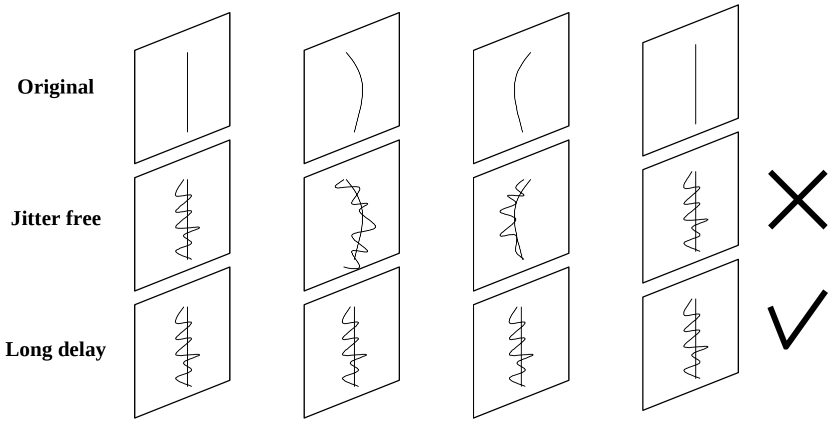

Figure 6 A qualitative illustration of the condition when delayed video looks better.

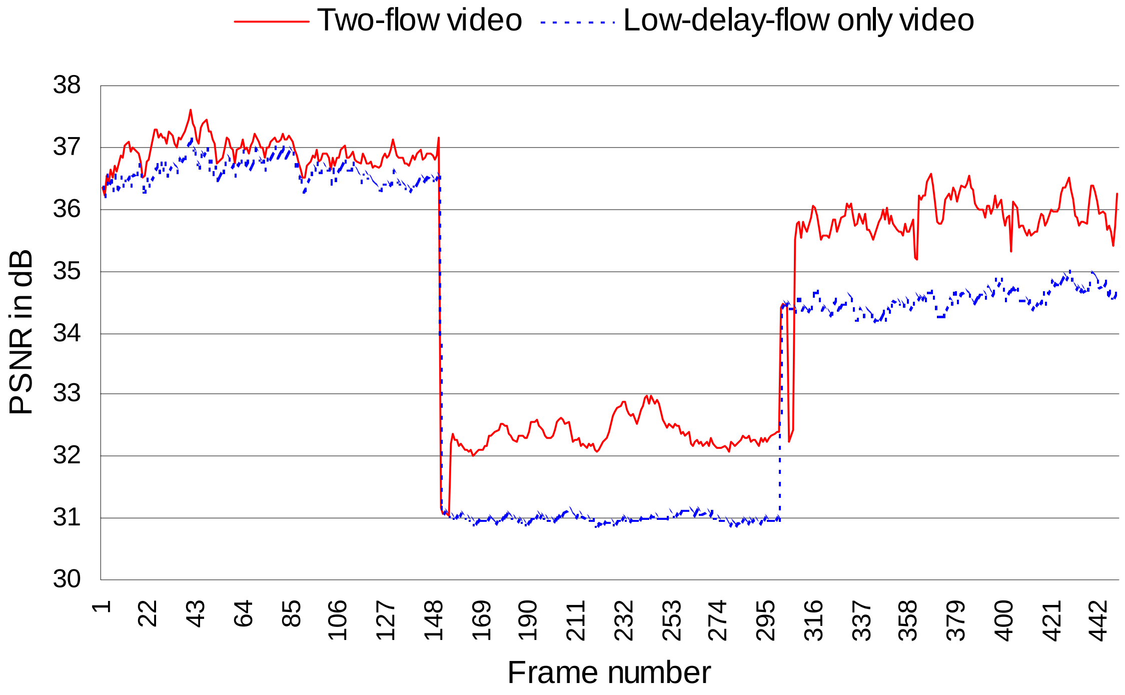

Figure 7 PSNR comparison in video quality of the low-delay-flow-only vs. DCVC two-flow video sequences.

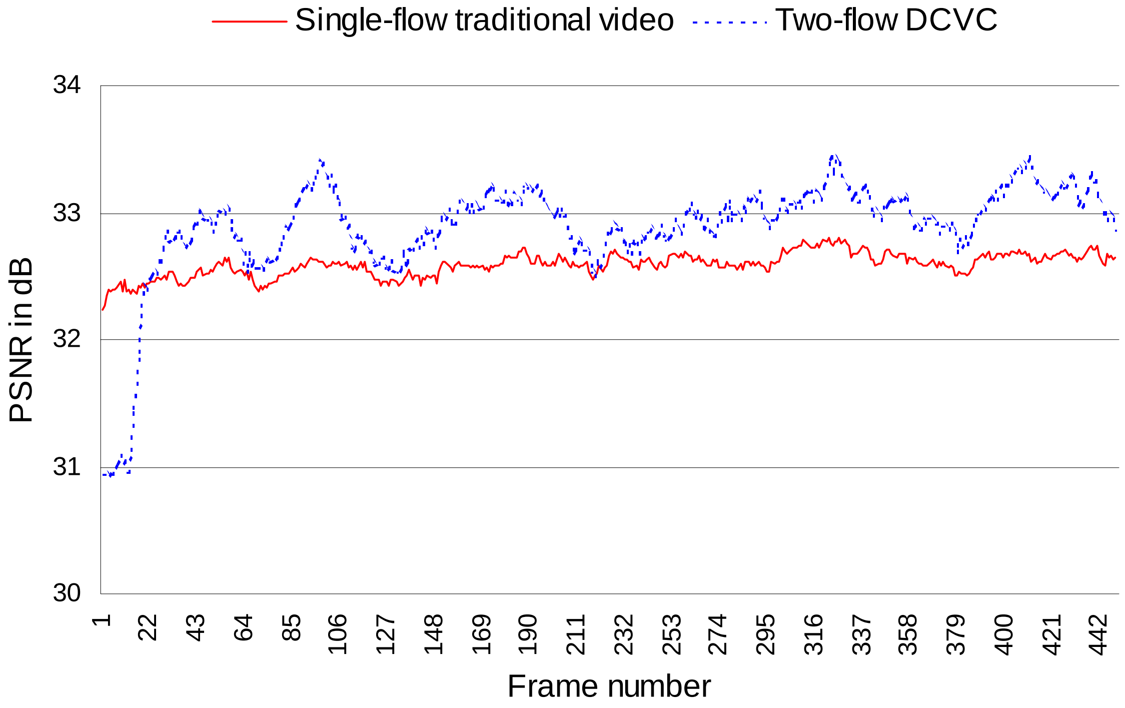

Figure 8 PSNR plots of H.263 and DCVC video to show comparable quality with 30% less bandwidth saving for DCVC.