Delay Cognizant Video Coding

Yuan-Chi Chang[1], David G. Messerschmitt[1], and Huen Joo Lee[2]

[1]Department of Electrical Engineering and Computer Science,
231 Cory Hall, University of California, Berkeley,
Berkeley, CA 94720, USA
[2]Information Technology Lab., LG Electronics Research Center, Seoul, Korea

Abstract

We address the coding of video using a substream model of transport. Joint source-channel coding (JSCC) is achieved in the coder by segmenting data into QoS substreams, and in the transport by provisioning different QoS attributes for those substreams. This paper focuses on JSCC in the delay dimension, which we call delay-cognizant video coding (DCVC), with the goal of increasing traffic capacity in the transport through traffic smoothing. A DCVC segments data by delay objectives and reconstructs the signal asynchronously at the receiver. Compared to conventional synchronous decoding, subjective quality is enhanced by making the perceptual delay representative of the minimum, rather than the maximum, transport delay. We discuss the motivation, design, and implementation of a delay cognizant video codec. Empirical quantification of the substream delay tolerance using perceptual distortion measures suggests that substream delay variability of ten frames is feasible for the video sequences studied.

1.0 Introduction

Untethered (and hence wireless) access to video and other multimedia services is desirable. This paper addresses the problem of achieving high traffic capacity in heterogeneous networks with wireless access subnets. We particularly focus on two issues: achieving high traffic capacity on the wireless subnet through joint source-channel coding (JSCC), and achieving perceptually low delay for interactive services (such as video conferencing).

Video coding cannot be designed too specifically for the wireless medium, because it is being transported through two or more subnets. For example, MPEG-2 (targeted at storage and high-reliability backbones) via wireless multi-access subnets is inefficient since the bit stream has to be heavily error-protected or repeatedly retransmitted.

In [1] we proposed a framework for a heterogeneous multimedia network, addressing the problem of JSCC by abstracting the transport as substreams with distinct QoS attributes, and segmenting the source data into these substreams in accordance with high subjective quality and minimal traffic impact. We call this QoS-centric coding to distinguish it from medium-centric coding, where specific knowledge of the medium is embedded in the source coder. QoS-centric coding can be performed in several dimensions as shown in Table 1, including rate, loss/corruption, and delay or combinations thereof.

Here we focus on delay, first studied in [2][3], and call this a *delay cognizant video coder (DCVC)*. A DCVC segments its data into substreams a different delay attributes. A DCVC decoder may reconstruct video asynchronously, since the substreams deliberately have distinct delay attributes and we may wish to avoid the artificial delays of resynchronization. In this case, to distinguish this coding from the traditional synchronous frame-by-frame reconstruction, we also call this asynchronous video (ASV) coding [2][3].

DCVC can indirectly result in an increase in traffic capacity, since the transport can apply traffic smoothing. For example, a wireless media-access layer has greater flexibility as to when to transmit packets for less delay-stringent substreams, allowing it to avoid the worse periods of interference. Our DCVC design goal is to make the *perceptual* delay representative of the *lowest*-delay substream, resulting in a reduction in perceived delay for a given transport delay profile, and allowing us in turn to relax the worst-case transport delay and increase traffic capacity further.

This paper describes a specific DCVC design, and reports experimental results as to its performance. In particular, we are concerned with the fundamental question of what transport delay profiles are permissible; that is, how much can the maximum delay deviate from

This research was supported by an IBM Cooperative Fellowship, the Advanced Research Projects Agency Contract J-FBI-93-153, LG Electronics, and Tektronix.

the minimum delay. Our particular design is block-based, and segregates blocks by their estimated motion and texture into three substreams: *low, medium*, and *high* delay. The low (high) delay substream carries the most (least) visually significant information.

2.0 Delay cognizant video coding

2.1 Transport abstraction

DCVC assumes a "medley" architecture [1], which is our name for a transport service that provides "flows" or "substreams". For purposes of DCVC design, we are primarily interested in the delay attributes of the substreams, which are assumed different. Ways in which the transport can provision such services, and take advantage of the delay differentiation, are beyond the scope of the paper. If the source coder generates the low-delay substreams as little as possible, while preserving subjective quality, we have achieved JSCC in the delay dimension.

Much recent video coding research has emphasized rate scalability and error resiliency [4]-[8]. A typical approach is to segregate the information into layers, which can also be transported through substreams. The use of substreams for JSCC in delay is a new contribution of our group [2][3], of which this work is a continuation.

2.2 Traffic efficiency improvement

In a time-varying wireless network, it is advantageous to schedule packet transmissions at advantageous times; for example when there is lower interference or to avoid fading events. Figure 1 illustrates this qualitatively, where a conventional and DCVC algorithm generate the same traffic, segmented into three substreams in the case of DCVC. We assume the low delay substream cannot be delayed, whereas the other two substreams have relaxed delay bounds. On the time-varying wireless channel, packets are dropped or delayed when there is insufficient instantaneous resources, such as bandwidth, transmit power, or received SNR. DCVC enhances the capacity because selected packets can be transmitted after the channel returns to normal.

DCVC is also intended to couple perceptual delay in interactive applications to that of the lowest-delay substream, allowing us to relax the delay of other packets. Because the delay is no longer fixed, we must substitute a subjective measure of delay. The idea is to separate the data most important to perceptual delay, namely blocks of the video frame with the highest motion content, and sending it via the lowest-delay substream. As we will see, the perceptual delay can be considerably smaller than the worst-case transport delay.

3.0 Codec architecture

In the current implementation, video blocks are segmented into substreams primarily by motion content (high-motion blocks are more sensitive to delay than low-motion blocks). After segregation by motion (no, low and high motion), block coding (such as vector or transform) is used to obtain a compact representation before transport by the appropriate substream. The decoder reverses the transform coding to obtain the original block representation, and block is displayed asynchronously (at time of arrival). The decoder does not wait for every block in a frame to arrive, and thus displaces blocks relative to one another.

By itself, asynchronous reconstruction imposes little constraint on the block coding. Our approach is based on the InfoPad coder [9], namely vector quantization (VQ) for hardware simplicity and low power consumption. The DCVC consists of five primary functional blocks: texture estimation, motion estimation, rate monitoring, VQ codebook search, and fuzzy control. As shown in Figure 2, an incoming video frame is first parsed as 8x8 macroblocks, sent to texture and motion estimators, which are fuzzified and pooled for the fuzzy controller. Based on predefined fuzzy rules, the fuzzy controller inference engine decides, for each macroblock, which codebook and which substream to use. The VQ coding algorithm then divides the macroblock into four 4x4 blocks and searches the assigned codebook for the best matched codevectors. The codevector indexes are packetized, labeled with a frame number and time stamp, and sent to the assigned substream. (The frame number is used to detect stale data at the decoder, and the time stamp allows any scheduler within the transport to infer the upstream transport delay.)

TABLE 1. Examples of QoS centric coding

QoS centric video coding	QoS parameter	Coding technique
Error resilient video coding	Loss/corruption	Unequal error protection
Rate scalable video coding	Rate	Multi-rate, multi-resolution
Delay cognizant video coding	Delay	Unequal delay requirement

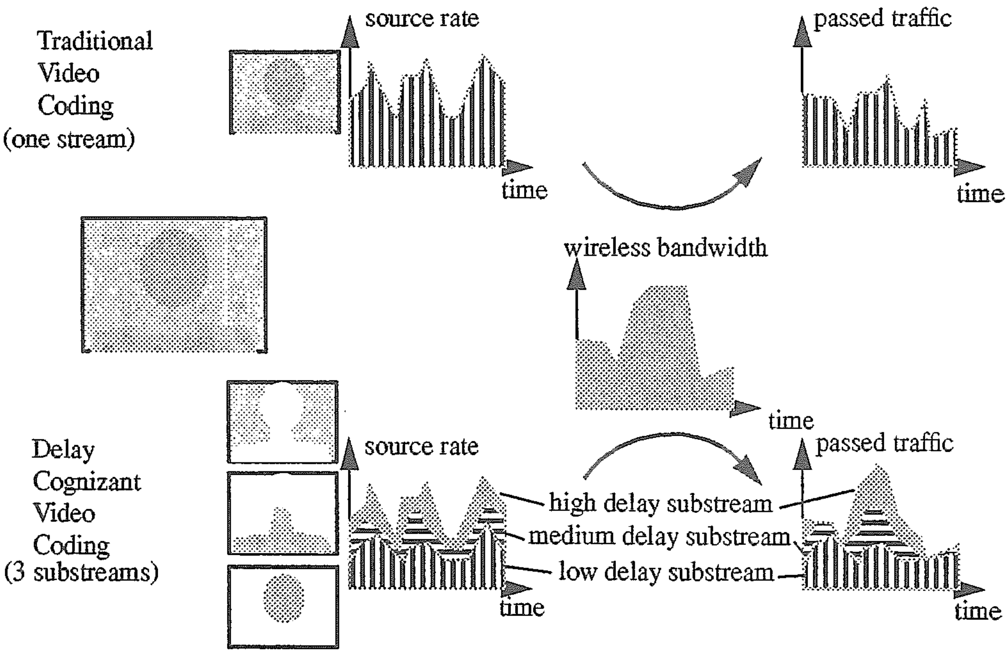

FIGURE 1. An illustration of the increase of traffic capacity by making the video coding delay-cognizant.

The rate monitor informs the controller whenever the pre-agreed traffic parameters are about to be violated, and also has presents a fuzzy interface to the controller.

Texture estimation of macroblocks improves the coding efficiency by adjusting the VQ codebooks to different resolutions based on texture as well as rate. For example, uniform areas require fewer bits (smaller codebook) while contour areas require a larger code-

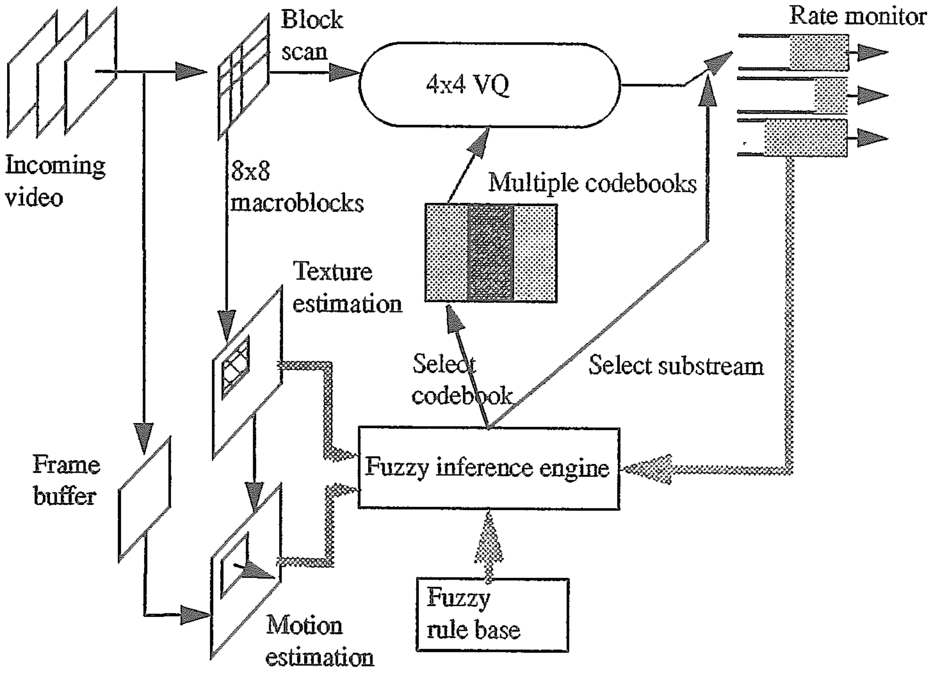

FIGURE 2. Encoder architecture

book. Quality is improved, and rate is reduced.

The motion estimation performs block matching with the previous frame. If a similar block is found near the current position, it is considered as a low- or no-motion macroblock. If a similar block cannot be found, high motion content is assumed.

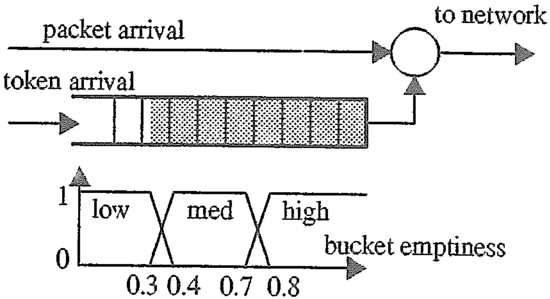

FIGURE 3. Fuzzification mapping of token bucket emptiness to fuzzy properties. Shown here is the membership function and its thresholds.

Feedback from the rate monitor forces an adjustment in output bit rate if warranted at substream granularity. Unlike passive traffic shaping, which drops or delays packets arbitrarily, this active traffic shaping allows the coder to make better use of bandwidth by transmitting the most visually significant information when the bandwidth is restricted. The rate monitor observes the output leaky bucket empty level, as shown in Figure 3. If the level is high, it asks the encoder to slow down. Each substream is assigned a leaky bucket monitor. The feedback is actually a set of fuzzy parameters, allowing the encoder to gradually change its coding decisions and thereby obtaining a smooth transition in quality.

We designed VQ codebooks with three different sizes -- 128, 256, and 512 -- based on classified vector quantization (CVQ)[10] as described in [12]. CVQ uses a front end classifier based on certain features, and the codebook search starts with the range tailored to these features. Codebooks with different resolutions result in better quality by coding the macroblocks with low motion and contour texture with high resolution, as quantization of these blocks is more visually significant. This also provides another dimension to rate control.

The fuzzy controller is the most interesting aspect of the design. It pools feedback from texture and motion estimators and rate monitor, and then makes decisions as to codebook and substream for each macroblock. The motivation for fuzzy logic twofold. First, fuzzy descriptions for block attributes and rate provide a structured and unified approach to construct a knowledge-based coder. Second, the fuzzy interfaces between the controller and the estimators provide an effective modularity.

The fuzzy controller inference engine uses a rule base to decide which codebook and substream to use. *Max-Min* rule composition is used as the inference procedure and the fuzzy rules are expressed as IF-THEN statements. First, the inference engine takes the *minimum* of the premises in the IF statement, which gives the similarity level of the consequent. Among all active rules, the consequent that has the *maximum* value is the coding decision. As a simple example, suppose the following two rules are active.

1. If (the block goes to the medium delay substream = 1.0) AND (its texture content is medium = 0.8) AND (high bit rate is allowed = 1.0), THEN (use the 512 codebook = min(1.0, 0.8, 1.0)).
2. If (the block goes to the medium delay substream = 1.0) AND (its texture content is high = 0.2) AND (high bit rate is allowed = 1.0), THEN (use the 256 codebook = min(1.0, 0.2, 1.0)).

The consequent of the first rule has value 0.8 while that of the second rule has value 0.2. Therefore, the first rule is chosen and the 512 codebook is chosen.

The design principles of this knowledge-based rule base can be summarized as follows:

1. Macroblocks of low delay substreams are coded with equal or lower resolution than those of higher-delay substreams.
2. Macroblocks with medium texture (usually representing contours) are coded with equal or higher resolution than those with high or low texture.
3. Macroblocks are coded with equal or higher resolution when more bandwidth is available.

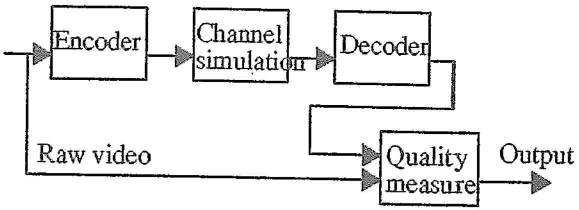

FIGURE 4. A schematic diagram of the evaluation setting.

As opposed to the complexities of the encoder, the decoder structure is simple. Its main function is finite-state machine and a table lookup. For each block received, the decoder checks its frame number and compares this number with the frame number of the block being displayed. If the received block comes from a later frame, it replaces the one in the frame buffer. Otherwise, it is dropped as it is *stale* (out of date).

4.0 Quantification of delay tolerance

In evaluating the effectiveness of DCVC, the most important performance indicator is the trade-off between subjective quality and delay variation among substreams. If only a small delay variation is permissible, not much is gained in transport traffic capacity. In fact, we will now estimate the allowed delay variation to be substantial.

4.1 Methodology

A schematic diagram of the empirical evaluation is shown in Figure 4. Test video sequences were coded, passed through a channel simulation, decoded, and compared with the original. Software that attempts to quantify subjective impairments was used as the basis of comparison. There are two sources of visual artifacts: compression and asynchronous reconstruction. The quality measurement tool applied to only the luminance component, so only monochromatic video sequences were evaluated.

The channel simulation was very simple: it delayed encoded data artificially to model differential substream delay. Every video packet in a substream was delayed by the same fixed amount referenced to the low delay substream. The minimum increment in delay is the frame display time, 1/30 second.

It is difficult to quantify perceptual delay, a major focus of our coding algorithm. To our knowledge, no prior research has created a metric incorporating both image quality and perceptual delay. Nevertheless, we hesitate to use the PSNR, for it does not to correlate well with subjective judgements. Rather, we used a measurement tool modeled after human visual systems, although it too does not quantify perceptual delay. It was developed by Prof. Murat Kunt and Christian J. van den Branden Lambrecht of the Swiss Federal Institute of Technology, Lausanne, Switzerland. Their work took

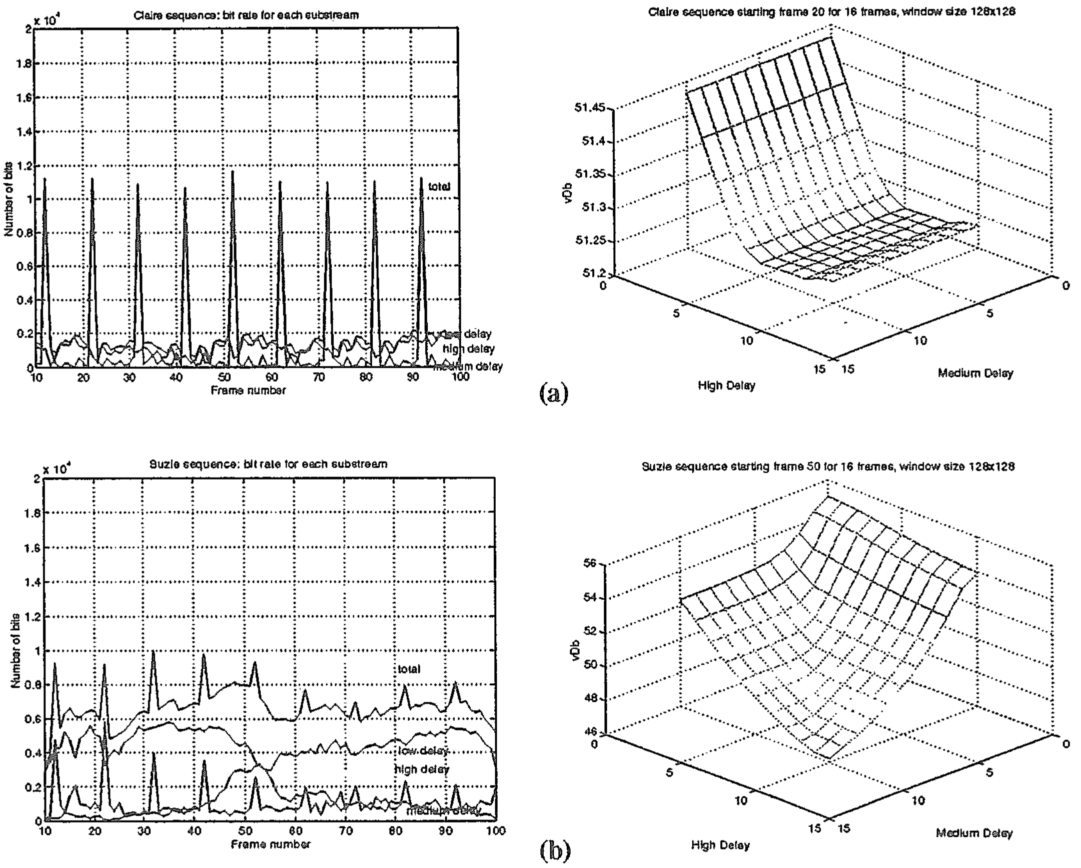

FIGURE 5. (a) Claire sequence; (b) Suzie sequence.

a weighted noise approach by taking into account the human vision characteristics. The evaluation is modeled after the multi-channel structure of human vision and accounts for contrast sensitivity and masking effects [17][18].

4.2 Results

Four test video sequences were used: Carphone, Miss America, Claire and Suzie. Each was in QCIF format with a resolution of 144x176. Due to limited space, we describe only the results of Claire and Suzie in Figure 5. These are mainly head and shoulder scenes and the movement of the subject is minimal. The evaluations were made by comparing the delayed versions directly with the raw video.

The results for each sequence are presented in two figures. The first shows the bit rate of each substream on a frame-by-frame basis, and thus defines the portion of the total traffic that can be delayed. The overall compression ratio, from 25 to 40, results from vector quantization, low resolution codebooks, and intermittent transmissions of no-motion blocks (which are retransmitted every ten frames to refresh the screen).

The second figure plots the measured quality of 121 test cases in visual decibels (vdB), the weighted signal-to-noise ratio on a dB scale defined in the evaluation tool. Each test case is composed of 16 frames with a frame size of 128 by 128. The high delay substream has a delay range evaluated from 0 to 10 frames and so does the medium delay substream. Therefore, the total number of combinations is 11^2, or 121.

4.3 Discussion

First, note that both high delay and low delay substreams contribute fairly high portions of the traffic. This is partly because the test sequences are talking head scenes, which are composed of high motion head and lip movements as well as still background images. The spikes came from periodic refreshments of no-motion blocks every ten frames. Second, from the video quality plots, the dynamic range of measured quality is fairly small. According to the authors' informal subjective assessment, sequences with less than 3 vdB difference look almost the same. In the following discussions, the subjectively acceptable criterion will be based on this threshold. If a test case is within the 3 vdB range of the original, it is considered to be acceptable.

As expected, shorter delays result in better quality. Degradations contributed by the delay of the high delay substream is, in general, less than that by the delay of the medium delay substream. This is, however, not the case for Miss America and Claire. A careful examination of their bit rate plots reveals the reason. There are only a few or even no blocks in the medium-delay substream, which thus does not affect measured quality, and most are in the high-delay substream.

A typical visual artifact arising in asynchronous reconstruction originates with the misjudgment of the motion estimation algorithm. To the observer, motion is the movement of a foreground object, such as a person, a car, a plane, etc., and not the uncovering of a background region previously occupied by foreground objects. In our motion estimation, however, block searching is executed in the neighboring region by sim-

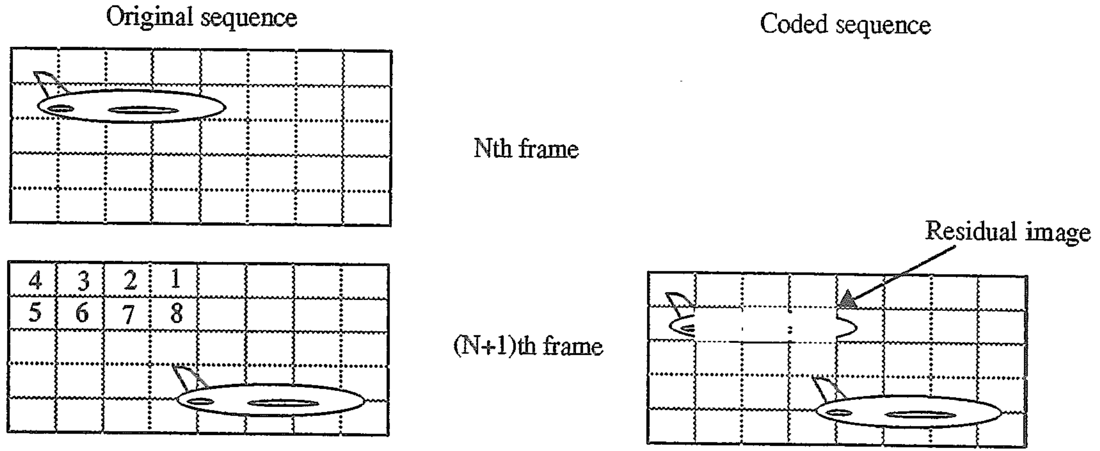

FIGURE 6. An example of visual artifact caused by misjudgments of the motion estimation algorithm; assume both high delay and medium delay substreams have nonzero delays.

ply comparing pixels, and does not distinguish whether the "motion" is a caused by a moving object or by the uncovering of a region. The situation is depicted in Figure 6. When the (N+1) th frame is coded, the motion search is done in the previous frame. As the airplane moves from the upper left corner to the lower right, macroblocks 4 to 8 are uncovered. Since the motion estimation does not know they are background blocks, it notes that similar blocks can be found a few pixels away. These macroblocks are mistakenly classified as low motion and sent through the medium delay substream. If blocks from the medium delay and high delay substreams are delayed, a residual image of the airplane will stay on the screen for a little while before it is cleaned up. Video sequences with medium to high motion content suffer this artifact most frequently.

Table 2 shows a the maximum delays that can be tolerated by each substream, in the units of frame, within the 3 vdB threshold. There are a number of feasible combinations and the numbers in the columns represent the largest *total* delay tolerance of both substreams.

The high delay substream can always tolerate 10 or more frames of delay (this is the maximum delay evaluated), or 330 msec. In video sequences with a lot of motion, the high delay traffic may represent a small portion of the total traffic, and thus the impact of this is reduced. However, interactive video applications like video conferencing rarely fall into this category; they are more likely to have head and shoulder scenes with still backgrounds, which generates a large number of no motion blocks that can be delayed.

The medium delay substream has a wide range of delay tolerance, from 2 to 10. The quality degradation mainly comes from the motion misjudgment described earlier. This may be acceptable in some less quality-critical applications. For quality-critical video applications, the medium delay substream can lag the low delay substream by no more than 2 frames. For non-quality critical applications, it can be delayed as many as 10 frames.

5.0 Conclusions and future work

We have presented a novel DCVC design. While the compression technology is straightforward, the design has several interesting features, such as multiple-substream rate feedback and a fuzzy inference engine controller. The primary novelty is segmentation of delay into delay sensitivity classes and asynchronous reconstruction.

Our major finding is that blocks with little or no motion can be delayed in transport for a long period (> 10 frames) without significantly affecting subjective quality or perceptual delay. Artifacts due to motion-based classification cause residual image traces when blocks with medium to high motion content are delayed.

We are currently exploring delay segmentation based on temporal subband coding. This should allow us to eliminate blocks, and block artifacts with them. We hope to delay the low temporal frequency components without significantly affecting subjective quality. The choice of temporal subbands and their delay tolerance are the main issues yet to be answered.

TABLE 2. Delay tolerance of the medium and high delay substreams

Sequence Name	Delay Tolerance (frame time), MAX(Med+High)	
	Medium Delay	High Delay
Carphone	10	10
Miss America	10	10
Claire	10	10
Suzie	2	10

6.0 Acknowledgment

Junjing Yan contributed to aspects of the design of the VQ DCVC, and also to the software coding, in her Master's project. Professor Murat Kunt and Christian J. van den Branden Lambrecht of the Swiss Federal Institute of Technology, Switzerland provided the video quality measurement tool and helped us use it. The software framework of our codec was derived from the video server in the InfoPad project, which was developed by Jeff Gilbert, A. Chris Long, and the Berkeley Plateau Multimedia Research Group.

7.0 References

[1] P. Haskell and D. G. Messerschmitt, "In favor of an enhanced network interface for multimedia services", IEEE Multimedia Magazine, to appear.

[2] D. G. Messerschmitt, J. M. Reason, and A. Y. Lao, "Asynchronous video coding for wireless transport", Proceedings IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA., pp. 138-45, Dec. 1994.

[3] J. M. Reason, et al. "Asynchronous video: coordinated video coding and transport for heterogeneous networks with wireless access", Mobile Computing, H. F. Korth and T. Imielinski, Ed., Kluwer Academic Press, Boston, MA., 1995.

[4] D. Taubman and A. Zakhor, "Highly scalable, low-delay video compression", Proc. ICIP-94, Austin, TX., vol. 1, pp. 740-4, 1994.

[5] W. Chung, et al. "A new approach to scalable video coding", Proc. Data Compression Conference, Snowbird, UT., pp. 381-90, 1995.

[6] E. Ayanoglu, et al. "Video transport in wireless ATM", Proc. ICIP-95, Washington, DC., vol. 3, pp. 400-3, 1995.

[7] H. Megal, et al. "A robust error resilient video compression algorithm", IEEE MILCOM, Fort Monmouth, NJ., vol. 1, pp. 247-51, 1994.

[8] A. Hung and T. Meng, "Error resilient pyramid vector quantization for image compression", Proc. ICIP-94, Austin, TX., vol. 1, pp. 583-7, 1994.

[9] S. Narayannaswamy, et al. "A low-power, lightweight unit to provide ubiquitous information access application and network support for InfoPad", IEEE Personal Communications, vol. 3, no. 2, pp. 4-17, April 1996.

[10] A. Gersho and R. M. Gray, "Vector quantization and signal compression", Kluwer Academic Press, Boston, MA., 1992.

[11] J. Lim, "Two-dimensional signal and image processing", Prentice Hall, 1990.

[12] K. N. Ngan, H. C. Koh, "Predictive classified vector quantization", IEEE Trans. on Image Processing, vol. 1, no. 3, pp. 269-280, July 1992.

[13] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial", IEEE Proceedings, vol. 83, no. 3, pp. 345-377, March 1995.

[14] B. Kosko, "Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence", Prentice Hall, 1992.

[15] A. Leone, et al. "An H.261-compatible fuzzy-controlled coder for videophone sequences", Proceedings of the third IEEE conference on fuzzy systems, vol. 1, pp. 244-8, 1994.

[16] S. H. Supangkat, K. Murakami, "Quantity control for JPEG image data compression using fuzzy control algorithm", IEEE Trans. on Consumer Electronics, vol. 41, no. 1, pp. 42-48, Feb. 1995.

[17] C. Lambrecht, O. Verscheure, "Perceptual quality measure using a spatio-temporal model of the human visual system", Proc. SPIE, Multimedia Computing and Networking, 1996.

[18] C. Lambrecht, "A working spatio-temporal model of the human visual system for image restoration and quality assessment applications", Proc. ICASSP, 1996.