Effective Algorithms for Video Transmission over Wireless Channels.

Pankaj Batra [1] and Shih-Fu Chang

Department of Electrical Engineering
Columbia University, New York, NY 10027
{pbatra,sfchang}@ctr.columbia.edu

Abstract

New schemes for video transmission over wireless channels are described. Content based approaches for video segmentation and associated resource allocation are proposed. We argue that for transport over wireless channels, different video content requires different form of resource allocation. Joint source/channel coding techniques (particularly content/data dependent FEC/ARQ schemes) are used to do adaptive resource allocation after segmentation. FEC based schemes are used to provide class dependent error robustness and a modified ARQ technique is used to provide constrained delay and loss. The approach is compatible with video coding standards such as H.261 and H.263. We use frame-type (extent of intra coding), scene changes, and motion based procedures to provide finer level of control for data segmentation in addition to standard headers and data-type (motion vectors, low and high frequency DCTs) based segmentation. An experimental simulation platform is used to test the objective (SNR) and subjective effectiveness of proposed algorithms. The FEC schemes improve both objective and subjective video quality significantly. An experimental study on the applicability of selective repeat ARQ for one way real time video applications is also presented. We study constraints of using ARQ under display constraints, limited buffering requirements and small initial startups. The proposed ARQ schemes greatly reduce the packet errors, when used along with optimal decoder buffer control and source interleaving. The common theme integrating the study of FEC and ARQ algorithms is the content based resource allocation for wireless video transport.

Keywords: Wireless video, mobile-multimedia, content or object based video coding and segmentation, content-based video transport

1 INTRODUCTION

Recently, there has been a great demand for audio/visual services to be provided over wireless links. However, due to bandwidth constraints, high error rates and time varying nature of

[1] corresponding author

these channels, the received video quality is still inadequate. Video transport over radio channels still remains an unsolved problem. New algorithms are needed to enable reliable video communication in wireless environments.

This paper describes new schemes for video transmission over wireless channels. It can be broadly divided into two parts. Firstly, content based approaches for video segmentation and associated resource allocation are proposed. Secondly, joint source/channel coding techniques (in particular, content/data dependent FEC/ARQ schemes) are used to do adaptive resource allocation. FEC based schemes are used to provide class dependent error robustness and a modified ARQ technique is used to provide constrained delay and loss. The approach is compatible with existing video coding standards such as H.261 and H.263 at rates suitable for wireless communication.

Traditional methods of segmenting video into substreams involve frame type, headers, and data type (e.g., motion vectors vs. transform coefficients). Improvement has also been shown by separating video to substreams and using adaptive resource allocation mechanisms [36]. These techniques are developed based on the notion that different types of video substreams have different levels of importance and should be handled differently. However, these approaches are restricted and do not take into account the "content" of the video. In this paper, we propose content-based segmentation methods to augment the traditional approaches. Here, video content refers to the inherent visual features present in the video. Examples are structures of the scenes, objects contained in the scene, attributes of the objects (e.g., motion, size, number etc.). Our goal is to demonstrate the value of the content-based video transport framework by providing proof-of-concept results of selected types of video content. We present resource allocation algorithms based on two simple types of video content: scene changes and extent of motion or activity[2]. Both of these features can be extracted from compressed video streams using automatic algorithms [27]. Our approach uses content processing schemes to classify the video content types and then allocate network resources (i.e., FEC and ARQ) adaptively.

An experimental simulation platform is used to test the objective (PSNR) and subjective effectiveness of proposed algorithms. A logical level figure illustrating the system architecture is given below (Figure 1). The first half of the paper involves the adaptive schemes based on FEC control. These algorithms are applicable to both one-way and two-way real-time applications. In the second half of the paper, we present algorithms using selective repeat ARQ for one-way real time video applications (like video on demand). The ARQ-based algorithms may incur retransmission delay which might be too long for two-way interactive services (such as video conferencing). To accommodate the variable delay caused by the selective use of ARQ schemes, our approach also includes an "elastic" buffer before the decoder/display module to "absorb" the delay jitter. The buffer absorbs the rate variation caused by selective ARQs and provide a compatible interface to the decoder. We will present efficient algorithms for buffer control in this case.

[2] For example, new frames after a scene change are subjectively important in order to establish the new visual context during the playback session. Image frames with high motion are less sensitive to errors due to the masking effects in the human vision model.

The remainder of this paper is organized as follows. Section 2 reviews existing approaches and related work in this area. Section 3 explains our proposed approach. Section 4 presents the schemes adopted for FEC. We also elaborate on the simulation method and results for forward error correction in this section. ARQ based recovery methods are explained in Section 5. Section 6 briefly summarizes the contributions of the work. We conclude in Section 7 by summarizing the current work and describing future directions.

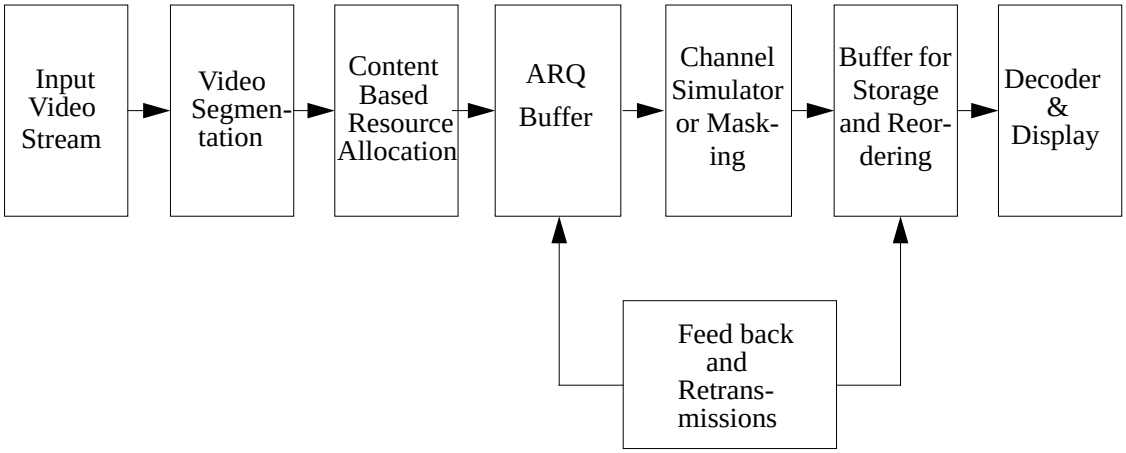


Fig. 1. *Logical Level Diagram of the System.*

2 RELATED WORK

Our work focuses on content based segmentation and associated resource allocation using modified ARQ and FEC schemes. Although there has been a lot of work in scalability, layered coding, and ARQ/FEC usage for videophone applications; the idea of content and object based approach to video coding is very recent and is largely limited to standardization bodies or related works [29,30,34,40]. Below, we give a brief overview of scalability, ARQ/FEC schemes and concealment options with particular emphasis on wireless video applications.

2.1 Data partitioning and layered coding

Scalable and joint source/channel coding are important ways for providing real time transport. Very early evidence of combined source channel coding technique exists in [28] which discusses its usage for images. More recently, works presented in [8,22,35] discussed the usage of such

techniques for real time services. Particular emphasis is given to scalable coding for VBR video over ATM networks in [9,12].

Data loss is a major problem for transmitting video over wireless networks. Bit error rates of around 10^{-3} to 10^{-2} may arise for long periods of time (approximately hundreds of milliseconds) especially in mobile environments. Data loss can be prevented if we used sufficient error correction codes, however, with increased bandwidth requirements. Further, errors normally occur in bursts and thus a worst case estimate of error correction overhead is not advisable. Thus, it becomes all the more necessary to use error protection judiciously. Unequal error protection schemes are also known to give graceful degradation over channels of varying quality [5,26] .

Some conventional data partitioning schemes are given in [26] and effect of transmission errors is studied for H.263 codecs. PET (or priority encoded transmission) related work at ICSI, Berkeley, is proposed in [25] and it uses a frame based segmentation/transmission approach. Its performance was studied for MPEG-1 Internet packet video. Arvind, Civanlar, and Reibman [2,3] study the packet loss resilience of MPEG-2 scalability profiles. They conclude that spatial scalability performs the best, followed by SNR scalability and data partitioning in that order. They consider cell loss ratios of about 10^{-3} [3]. But, burst error effects are not taken into account in that work. Amir and McCanne [1] have proposed a *layered DCT coder* which is derived from progressive JPEG. The coder defines a scalable structure by dividing the bit plane into layers. Asynchronous video coding schemes presented in [36] use conditional replenishment based schemes to discard fine-resolution information (e.g., high frequency DCT coefficients). This work will incorporate other known compression schemes such as vector quantization and motion compensation. Wavelet, subband [4,38], and pyramid coding techniques which lend themselves to scalable architectures are also being presented.

2.2 ARQ/FEC schemes

There have been many recent studies on use of FEC and ARQ schemes for wireless video. Khansari et.al. [23] have used a combination of scalable (dual rate) source coding with hybrid (Type I) ARQ and FEC for transmission of QCIF resolution H.261 video over wireless channels. Their work concludes that usage of a mixture of FEC and ARQ based structure needs less overhead as compared to usage of FEC alone for similar visual quality. Work at the University of Southampton [17,39] also uses hybrid ARQ and unequal-FEC codes to improve robustness in transport of DCT-based compressed QCIF videophone sequences. Modestino et.al. [33] recently studied the design of forward error correction codes for video transmissin over ATM networks. Recent work by Han and Messerschmitt [16] provides a "leaky ARQ" scheme that successively improves the quality of delay-non-critical portions of graphics by sending more refined versions. It trades off quality (reliability) for delay for portions (e.g., menus) that should appear without significant delay. This work was, however, limited to "window based text/graphics" and did not

[3] which translates into a BER of $O(10^{-5})$

study the extra considerations due to video [4]. Data (re)ordering is yet another way in which prioritization for retransmission can be done. More important data is retransmitted prior to less important portions in this case.

2.3 Concealment options and robust codecs

Concealment related approaches exploit the spatio-temporal correlation in the bitstream. These methods mainly comprise of temporal replacement and motion-compensated temporal replacement, or related ways of interpolation [10,11,24]. Further, they can be applied either interactively between the source and the receiver or locally at the decoder. Temporal error localization can be done by having frequent I frames or intra coded slices and spatial localization of errors can be done by having rate based adaptive slice sizes. Zhang et.al. [41] present a review on error concealment techniques which exploit the spatio-temporal redundancies of the MPEG-2 standard [14]. They also study the interactions between the MPEG-2 systems and video layers, and suggest additional ways in which video quality can be improved in wireless ATM environments. Current *MPEG-4 standardization efforts* are also actively investigating concealment schemes like duplicate information, two-way decode with reversible VLC, and data-partitioning [31,34] . They are trying to add the ability to quickly resynchronize and localize errors in compressed video streams. System, syntax, and MUX level support will also be provided by work being done in MSDL Working Draft [29–31,34]. Our work has great synergy with the content based theme of MPEG-4.

3 AN APPROACH BASED ON VIDEO SUBSTREAM SEGMENTATION

Traditional data filtering/segmentation methods have exploited the hierarchy in the coding structure. Both the H.26x and MPEG-x suite of standards divide the coded data into many syntax layers. In MPEG-2 (ISO/IEC 13818-2 Committee Draft), for example, the entire sequence is divided into group of pictures (GOPs). Each GOP has a specified number of pictures/frames, and starts with an intra-coded I frame. I frames have no motion compensation performed on them. Also present are the P frames which are predictively coded from previous I and P frames. Between the anchor frames (I/P frames), are bi-directionally predicted B frames. Each picture is further composed of slices, which in turn consist of macroblocks. 16×16 macroblocks consist of 4 8×8 blocks. Motion compensation is applied at a macroblock level, wherein a best matching block is found and one or two motion vectors are sent in the bitstream.

Because of this hierarchy in coding, wireless errors affect video quality depending on where they hit. Errors occurring in headers (e.g., sequence startcodes, GOP headers, picture headers, slice headers) have the most detrimental effect. Next, the effect of errors depends on the image frame

[4] e.g., real time deadlines, buffer requirements, error propagation, and inter frame coding etc.

type – the sensitivity being I>P>B. The perceived error sensitivity for picture data is motion vectors > low frequency DCT coefficients > high frequency DCT coefficients. Depending on the above, we can define a data segmentation profile in decreasing order of priority in terms of error robustness. The above is basically the philosophy behind the MPEG-2 data-partitioning method.

For wireless applications, the H.26x suite of standards are more popular due to their focus on low bitrate. Although, there are additional options (e.g., PB frames, extended motion vector range etc.) in H.263, the underlying structure of codecs is very similar.

Our approach differs primarily in that it uses video content as a means of further data segmentation. Our goal is to develop a content-based video transport framework. Video "content" is quite general. But, in this paper we provide proof of concept by showing this new approach based on video segmentation based on motion and scene changes, and its interaction with existing video partitioning based on headers, and data-type. We use a psycho-visual (i.e., content based) framework for video segmentation. A typical video sequence can be divided into a set of independent scenes. Further, a scene is composed of frames of different activities (motion). We use scene changes and motion to further segment the video stream. For resource allocation, we argue that different video content requires different form of resources to be allocated to it. For example, new frames after a scene change are subjectively important in order to establish the new visual context during video playback. Image frames with high motion are less sensitive to errors due to the masking effects in the human vision model.

The primary QoS constraints which we consider are error robustness and bounded end-to-end delay. We use variable FEC to provide multi-tile error resilience. We study both the individual and aggregate effects of using video headers, frame-type, motion, scene changes, transform coefficients, and motion vectors as a basis for choosing the error correction overhead. We also use a restricted link level retransmission scheme to provide bounded end to end delay. We propose innovative schemes for selectively applying ARQ-based retransmission and controlling the interface buffer between the receiver and the video decoder.

4 FEC BASED SCHEMES

4.1 Frame Segmentation Based on Scene Changes or the Extent of Intra-Coding

We use the H.263 codec for encoding video. Hence, INTRA/INTER mode decisions for prediction are made at a macroblock level [15]. The scheme to make these decisions is the same as specified in TMN5 (Test Model Number 5). Given a pre-encoded video sequence, frames are prioritized based on the fraction of macroblocks that are intra-coded in them. Following this, three partitions are generated as:

$$\alpha = \frac{\#INTRA\_CODED\_MACROBLOCKS}{\#INTER\_CODED\_MACROBLOCKS+\#INTRA\_CODED\_MACROBLOCKS}$$

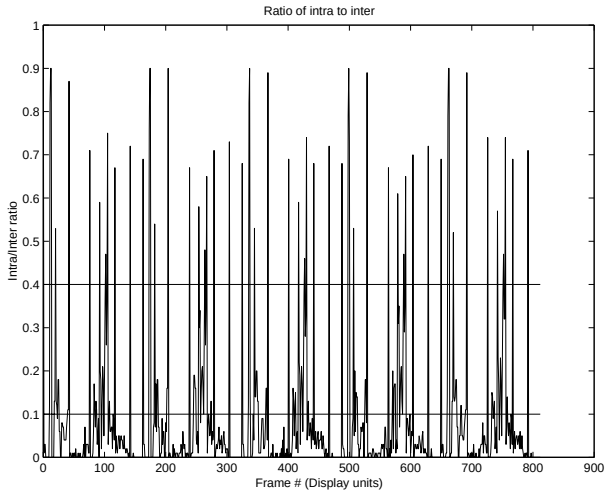


Fig. 2. *Percentage of Intra-Coded macroblocks in a typical H.263 video sequence (α)*

Partition 1: $\alpha \in [0.4, 1]$ Highest priority

Partition 2: $\alpha \in [0.1, 0.4)$ Medium priority

Partition 3: $\alpha \in [0, 0.1)$ Lowest priority

The choice of partitioning thresholds (i.e., 0.1 and 0.4) is adhoc here. Variations in α for the sequence under consideration are shown in Figure 2. The priority assignment we consider here is in terms of error resilience and is provided using variable FEC. The justification for using this ordering of priorities is that frames that are mostly INTRA coded provide refresh instants and help in preventing error propagation. Furthermore, perceptually too, scene change frames are important for a complete understanding of the new video scenes. Hence we would like to have least number of errors in them. These frames can be used to provide scene change based scalability in addition to other schemes of temporal scalabilility [5].

It is worth noticing that the first partition typically corresponds to the scene changes in the sequence. In the MPEG-1/2 video coding standards, partitions 2 and 3 usually contain P or B frames. In that, one could also obtain a two level partition for each frame, with the first level being the frame-type and the lower level partition based on the occurance or non-occurance of a scene-change.

4.2 Motion Based Segmentation

The segmentation is done in two levels – low and high motion. We estimate motion at a frame level. For each frame, a frame level average motion vector is calculated, and compared with the

[5] e.g., frame filtering options in MPEG which drop B or B+P frame combinations

global average. This is used to classify frames into high and low motion ones. Motion based segmentation can also be applied at a finer granularity e.g., at the macroblock or VOP [6] level. In terms of error robustness, it is known that a high motion layer can tolerate higher errors as compared to a lower motion layer due to masking effect [6,36]. Hence, we assign lower FEC overhead for high motion frames as compared to low motion ones.

4.3 FEC Based Resource Allocation

Figure 3 shows the hierarchy for segmentation for error-robustness. Each state (leaf node) corresponds to a given set of allocated resources. Nodes higher up in the tree have more priority than those below. Similarly, nodes to the left carry more priority than those to the right. We use stronger FEC protection for higher priority layers. Attributes having highest incremental effect on the video quality are placed higher up in the tree and to the left side. Headers form the highest priority layer to ensure that the corrupt bitstream can be decoded properly. This is followed by scene change frames (Partition 1 above). Frames that are not in partition 1 are further divided into partitions 2 and 3. Motion information is used to again split these partitions. Finally, the actual data (motion vectors and DCT coefficients) are at the innermost layer, and normal MPEG or H.26x data partitioning like schemes are applicable within the frame. Specific allocation schemes of different degrees of FEC and their experimental results will be presented in Section 4.4.

4.4 FEC Results

We use the wireless channel simulator described in [18] . The wireless channel is modeled using the Rayleigh fading multipath model. Personal Access Communications Services (PACS) system is used for the air-interface [32]. The Rayleigh fading channel is simulated using the Jakes Model [21] . The simulator is used to generate simulation data with errors under different error controls. The final data with errors is compared with the original data to generate error masks. Error masks are used to mask the video data to simulate the transmission of video over the wireless channel. The wireless channel simulator [18] is used to generate bit error pattern files on an off-line basis. Error pattern files can be generated for different channel conditions, error correction capabilities, and interleaving degrees. The coded video sequence is sequentially parsed, and depending on its current content, an appropriate error pattern file is used to mask it. Other wireless parameters used in the FEC-simulations are provided in Table 1 [7] .

The algorithm is implemented using H.263 codec. Further details on the video parameters used

[6] VOP or video object plane is a temporal instance of an arbitrarily shaped region in a frame. It is defined in MPEG-4.

[7] Courtesy: The authors thank Dr. Li-Fung Chang (Bellcore) for this information.

Maximum Doppler Frequency $f_d = 1Hz$.
Perfect carrier recovery.
Optimal symbol timing.
Transmitted signal power = 19db.
QPSK modulation (2 bits/symbol).
Coherent demodulation.
Matched Nyquist filter with a roll-off factor of 0.5
Diversity = 1.
Round trip delay = 6msec [32].
Multiple access = TDMA
Air Interface: 400 frames/sec each of eight 32kbps slots.
Code length = 40 symbols.
Symbol length = 64 bits.
FEC = RS(40,40-2t).
Interleaving degree = 40

Table 1
Wireless parameters used in the simulation [16].

Sequence = Unrestricted CNN news sequence (including commercials)
Codec = H.263 [15]
Rate control = Simple offline rate control (with fixed frame rate)
Rate control ON for the entire sequence
Resolution = QCIF (176x144 pixels)
Chroma format = 4:1:1
Motion Estimation Search window = 10 pels
Bit rate = Constant bit rate $\approx$ 32 kbps
Frame rate = 7.5 fps
All enhancement options ON
Number of frames $\approx$800 (106 sec.)
Other encoding specific parameters and schemes are as in TMN5

Table 2
Video parameters

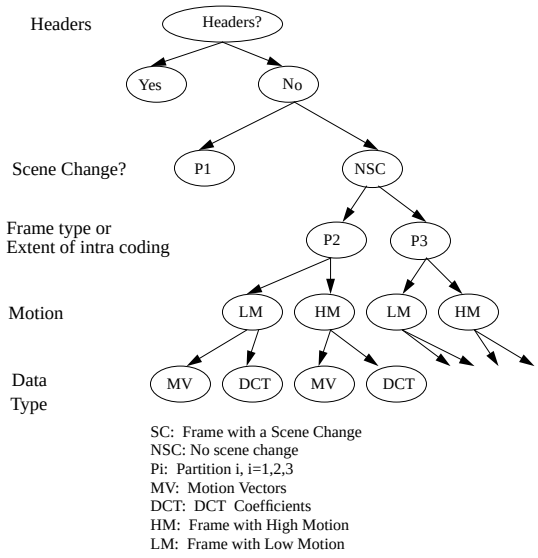


Fig. 3. *Template Assignment for Error Robustness.*

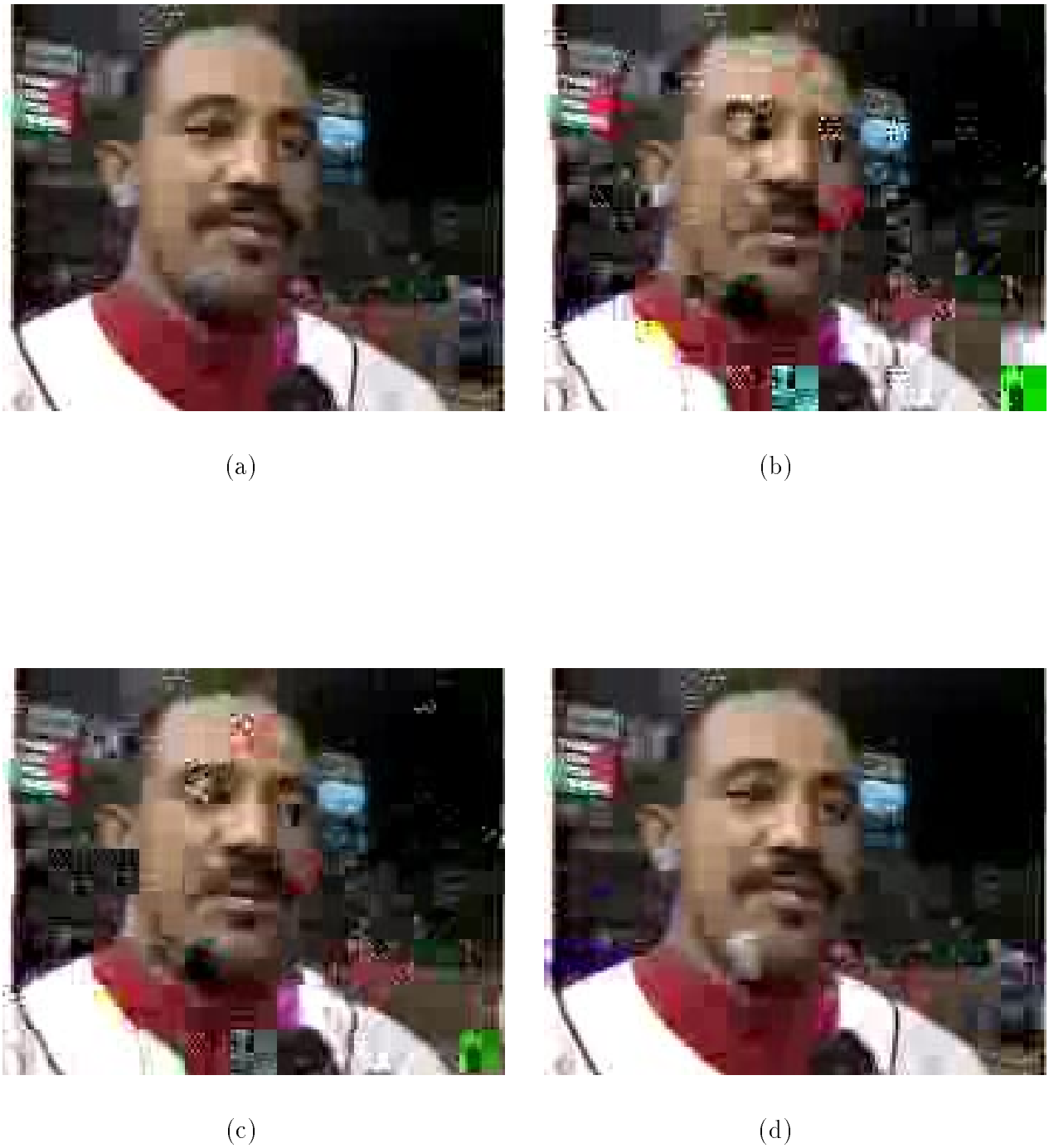
in the simulation are given in Table 2 . Quality of the final decoded video is measured using its PSNR, defined as [31]:

$$\text{PSNR} = 10 \log_{10} \left(\frac{1.5 \times 176 \times 144 \times 255^2}{\sum (Y_e - Y_d)^2 + \sum (U_e - U_d)^2 + \sum (V_e - V_d)^2} \right)$$

where the subscript e indicates the decoded version of a frame with errors and d stands for a normal decoded frame (not the frame that was encoded). Y is the luminance, and U and V are the chrominance components respectively. Further, because the proposed algorithm is psychovisual in nature, subjective evaluations are also made on the decoded stream.

In the following, we study the incremental effect of using headers, scene-changes, intra-coded macroblock percentage (α), and motion as a basis for choosing the error correction capability of the code. The PSNR graphs plotted below show PSNR between a normal decoded sequence in the absence of any channel errors and that with channel errors. The large spikes (bound by 150 db.) indicate infinite PSNR or no errors in decoded video. To maintain fairness in comparisons, average FEC overhead in each case was maintained approximately same. The total overhead due to channel coding is about 25% for all cases. The average frame level PSNR is compared for the various cases under consideration. Figure 4 shows results for a typical frame. Figure 4(a) is the normal decoded frame. Figure 4(b) shows the same frame after it is transmitted over the wireless channel. No data segmentation or concealment is used in Figure 4(b). Figure 4(c) shows the result for header plus α based segmentation. We see that there is a significant improvement in quality as compared to 4(b). Figure 4(d) uses motion information in addition

to headers and α based segmentation. The subjective quality improvement is quite obvious, especially in the impaired image areas.



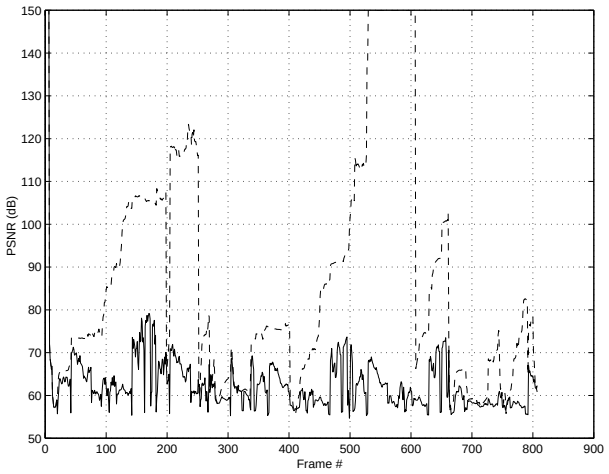
(a) (b)

(c) (d)

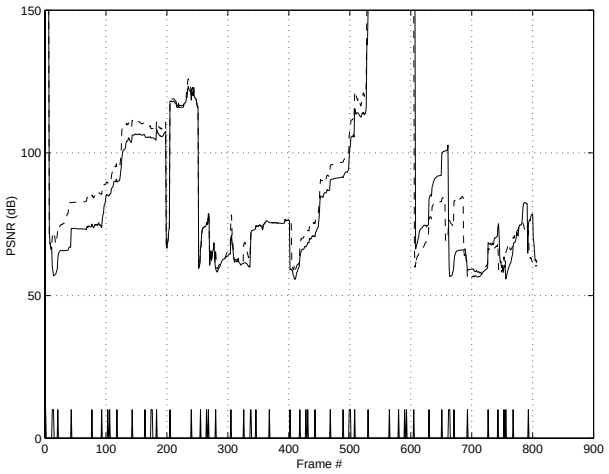
Fig. 4. *Results from different FEC allocation schemes. Channel BER without any FEC* $= 5.1 \times 10^{-3}$ *on the average. (a) Original Sequence (b) No data segmentation (c) Headers + P1 + P2 + P3 (d) Headers + P1 + P2 + P3 + motion*

Figure	Scenario	Parameters	$t_{effective}$
(a)(solid)	No Segmentation	t=5	t=5.00
(a)(dashed)	Header Based Segmentation	t(header) = 9, t=5 else	t=5.04
(b)(solid)	Header Based Segmentation	t(header) = 9, t =5 else	t=5.04
(b)(dashed)	Header + Scene changes	t(header) = 9, t(scn)=6,	t=5.21
	$\alpha \in [0.4, 1]$	t=5 else	
(c)(solid)	Same as (b)(dashed)	t(header) = 9, t(scn)=6,	t=5.21
		t=5 else	
(c)(dashed)	(b)(dashed) + P2 or $\alpha_{medium} \in [0.1, 0.4]$	t(header) = 9, t(scn)=6,	t=5.28
		$t(\alpha_{medium})$=6, t=5 else	
(d)(dashed)	High motion – lower FEC	t(header) = 9, t(high motion)=4,	t=5.06
		t(low motion)=5	
(d)(solid)	High motion – higher FEC	t(header) = 9, t(high motion)=6,	t=5.03
		t(low motion)=4	

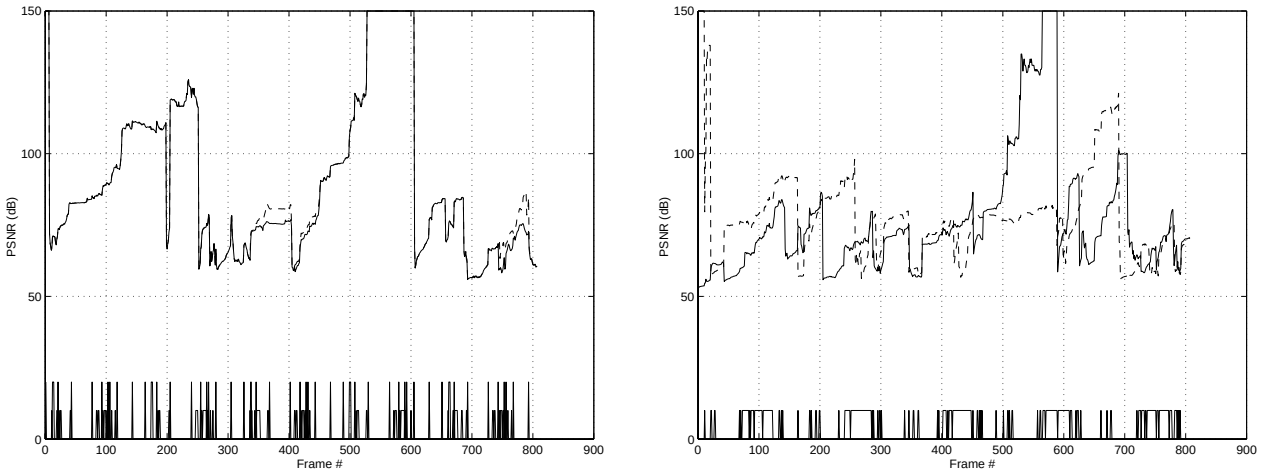
Table 3
Simulation conditions for FEC



(a)



(b)



(c) (d)

Fig. 5. *Incremental effects of various attributes, BER of the channel = 5.1 $\times$ 10^{-3}. The PSNR values shown are between a decoded frame with errors and that without any errors (and not between a decoded frame with errors and the original frames which are encoded). Note that scale in (a) is different from that in (b) onwards.*

FEC parameters for Figure 5 are shown in Table 3. t is the error correction capability of the code. The effective overhead for one of the curves in each of the graphs (a), (b) and (c) is higher than the other one. Thus, our objective here is to compare the unit increase in FEC overhead per unit increase in PSNR across graphs. Figure 5(a) compares PSNR of the decoded stream with no data segmentation and that with FEC done based on headers (e.g., headers for the picture layer). From Figure 5(a) we see that a significant gain can be obtained by giving high protection to headers. Giving higher error protection to headers ensures that the decoder finds clean start codes and parts of video do not become un-decodeable till next sync. Figure 5(b) shows the incremental effect of giving higher protection to Partition 1 frames. These frames are largely intra-coded, and are assigned next higher priority to prevent error propogation. Black spikes in figure 5(b) indicate the location of high-intra frames. It can be seen that the dashed line (header + intra) consistently outperforms or equals the solid one (header only) for a very little increase in the effective error correction overhead. Figure 5(c) compares the effect of using Partitions 2 and 3 for further data segmentation. Based on the α value, we use different t values for Partition 2 and 3 frames. The large black spikes indicate frames in partition 1 and the smaller spikes are frames in partition 2. The dashed line is seen to be better than the solid one for all the frames. It is worth noticing that with a very small extra FEC overhead for frames in partition 2, perceptual quality improvement can be obtained. However, the increase is not as much as obtained with headers or Partition 1. In other words, although the effective error correction overhead is increased in both (b) and (c) (as compared to (a)(dashed)), the increase in this overhead per unit increase in PSNR is much less with (b) as compared to (c). Similarly, this unit overhead is least with (a). Thus, we justify the arrangement of top three layers of the tree shown in Figure 3. Figure 5(d) shows how error correction overhead should be

distributed among frames with high and low motion (activity). Earlier work in [6,36] suggests that error sensitivity with high motion is less than that with low motion. We have indicated frames with high motion by black spikes in figure 5(d). We consider two cases: in the first, high motion frames are given less FEC overhead as compared to frames with low activity (dashed line); the other case (solid line) is the opposite. We notice that the quality is degraded less in the former case as compared to the latter; and therefore earlier results are substantiated. However, since the FEC allocation is heuristic in this case, PSNR may not be the right metric to measure quality.

Overall, it is seen that headers give the largest improvement in SNR for minimal increase in error correction overhead. Scene change and the extent of intra coding can be used to further improve the visual quality at small extra FEC overhead. Further, we justify that giving less FEC overhead to high motion frames gives less degradation in the perceived quality. Looking at the incremental improvement obtained by each of the above criteria, we verify the design of the layering structure shown in the tree in Figure 3 . Within each criterion (e.g., partitions 2 and 3 combined), the chosen left to right arrangement is seen to give best results. At a given depth in the tree, leaf nodes to the left are more important in terms error robustness. Hence, if the underlying network allows only a limited number of service classes, or to decrease the FEC encoding/decoding complexity, it may be feasible to cut down on classes bottom up in the tree. Bundling leaf nodes into groups from left to right and providing them same QoS guarantees is yet another possibility.

5 RETRANSMISSION BASED SCHEMES.

5.1 Model

The link between the first part (on FEC) of this paper and the second part (on ARQ) is the common theme using adaptive resource allocation according to video content. In this section, we present the content-based algorithms in relation to ARQ. We first discuss the ARQ model, present the content-based schemes based on scene change and motion, describe an innovative algorithm for solving the delay jitter problem caused by the proposed ARQ scheme, and finally present simulation results.

A simplified system model is shown in Figure 6. We consider transmission across a single last hop wireless channel. We model channel delay as a random variable (currently deterministic) with some fixed mean value. Incoming compressed stream is interleaved and FEC encoded before transmission through the simulated channel. The ARQ buffer at the sender stores frames that have been sent but not yet acknowledged. Its size is a function of the ARQ protocol being used and the window size[8]. Similarly, an ARQ buffer for reordering of frames is required at the receiver. The feedforward loop is for display rate adaptation and its functionality will be

[8] The maximum number of en-route frames at any instant.

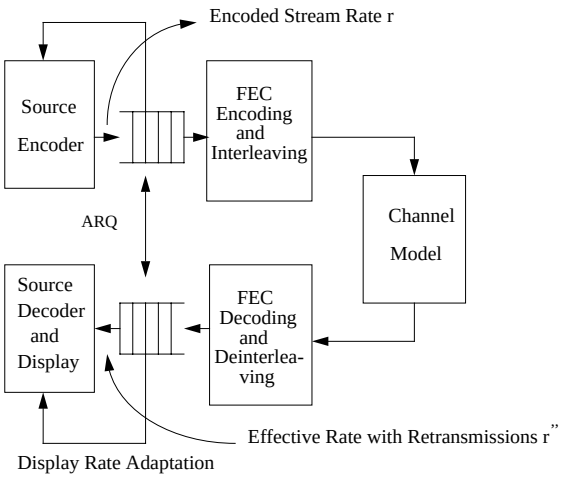


Fig. 6. *System Diagram of the proposed ARQ-based schemes. The buffers shown are ARQ buffers.*

discussed in later sections.

Video encoding parameters are indicated in Table 4. Unlike FEC simulations, the video is aggressively rate controlled using H.263 Online-rate control (See TMN5 for details). This H.263 option produces a variable frame rate bit stream with skipped frames so that target bitrate constraints are met. The approach differs from FEC simulations since our primary objective when studying ARQ is to contain decoder buffering requirements while meeting display constraints.

The optimal packet or frame size that should be used is in general a function of the data rate, channel rate, the ARQ protocol being used, the allowable latency etc. However, here we assume that a video frame is the packet entity under consideration. This gives us significant advantage in terms of speed, complexity, synchronization, packet handling and header integrity. Furthermore, for the above mentioned encoding parameters, the average video frame size is about 620 bytes which is a reasonably small size to be handled (ATM based transport uses a packet size of 188 bytes, which is of similar order of magnitude).

For 2-way interactive video, ARQ may not be suitable due to latency concerns. But, for 1-way video applications, the proposed scheme with retransmission as a last wireless hop recovery scheme can be used. Using ARQ for one way video playback is a complex task as shown in literature. We will elaborate more on issues regarding the usage of ARQ schemes in the following section. Experimental results justifying the approach are provided in later sections.

Sequence = Unrestricted CNN sequence
Codec = H.263 [15]
Rate control = Online rate control (with variable frame rate)
Rate control ON for the entire sequence
Resolution = QCIF (176x144 pixels)
Chroma format = 4:1:1
Motion estimation search window = 10 pels
Bit rate = Constant bit rate $\approx$ 24 kbps
Achieved frame rate = 6.4 fps (mean)
Enhancement options ON
frames$\approx$600 (about 94 sec)
Other encoding specific parameters and schemes are as in TMN5

Table 4
Video parameters for ARQ simulations

Class	Criteria: Intra and motion
C1	High Intra ($\alpha \in [0.4, 1.0]$) or High motion
C2	Otherwise

Table 5
Classes for ARQ

5.2 Segmentation

We segment frames based on two types of basic video content - the extent of intra-coding (α, equivalent to scene changes), and the extent of motion. Since using ARQ involves latency buildups leading to shifts in the display axis [9], the idea is to identify a subset of frames which are more important. Only this subset uses ARQ based recovery. High-intra frames prevent error propagation. Being scene changes, they are also important for full subjective understanding of the video. Hence they are allowed one retransmission for recovery in case they are in error. Motion is the second cue used in segmentation. High motion frames are allowed one retransmission for recovery since the content changes significantly from the previous display instant and most error concealment schemes at the decoder may not work. As illustrated in Table 5 , ARQ based recovery is used for frames of class C1 only and not for those belonging to class C2.

[9] These ideas are discussed in detail later.

5.3 Analysis for adaptive decoder buffer control

5.3.1 Issues

Due to selective ARQ, the CBR bit stream will suffer from rate variation at the receiver. Using the same notation as in [37] , the buffer evolution of the encoder buffer is given by:

$$B_i^e = B_{i-1}^e + E_i - R_i \tag{1}$$

with

$$0 \leq B_i^e \leq B_{\max}^e \tag{2}$$

where B_i^e is the encoder buffer occupancy at the end of frame period i, R_i is the channel rate for the same period, and E_i is the encoding rate for that period. The second constraint is required to prevent encoder buffer overflow or underflow. To prevent under/overflow of the encoder buffer, E_i and R_i should satisfy some constraints [10] . Rate control involves the proper selection of E_i (or equivalently the quantization parameter and frame-rate) and R_i so that these requirements are met.

Video transmission over networks normally assumes a fixed end to end delay between the encoder and decoder. The dynamics of the decoder buffer, is therefore, a time-shifted version of the above buffer dynamics, and is given by

$$B_i^d = B_{i-1}^d - E_i + R_{i+L} \tag{3}$$

with

$$0 \leq B_i^d \leq B_{\max}^d \tag{4}$$

where we assume an end-to-end delay of L frame periods and $B_o^d = \sum_{j=1}^{L} R_j =$ is the startup decoder buffering. For CBR video streams, it can be shown [37] that we can prevent underflow or overflow of the decoder buffer by preventing overflow or underflow at the encoder buffer. If the stream is VBR, then E_i and R_i must satisfy certain constraints to avoid buffer overflow and underflow in the encoder and the decoder [37].

The scenario we are considering is much more complex due to possible retransmissions. Hence, even if the video is aggressively rate controlled and the target constant bit rate is met, the display constraints may not be met. Or equivalently, the decoder buffer may underflow due to loss of time during retransmissions. Although it is possible to model retransmissions delays as simple network delay jitters and modify L to $L + \Delta$ [11] , the scheme has its drawbacks. Firstly, errors normally occur in bursts in wireless channels, and so the order of Δ we wish to dejitter

[10] Refer [37] for details

[11] This suggestion was made in [37] for channels having variable delay (i.e., with a jitter term added to a mean value).

is very large. This may require large initial startups or initial buffering to prevent underflow. Secondly, the channel characteristic is time varying and is not known apriori, and in general cannot be pre-negotiated (although channel dependent renegotiation is a possibility).

5.3.2 Analysis and Algorithm

The retransmission scheme we are using is a modification of the Go-back-W scheme with selective retransmission of erred packets. W, the window size, indicates the maximum allowed number of outstanding packets sent out by the sender which have not yet been acknowledged by the receiver. The receiver sends bundled ACK/NAKs at the end of the window and only frames in error are retransmitted.

We focus on a single retransmission period, i.e., W frames are sent and B of these are assumed to be in error and are retransmitted. Assume that t_{prop} is the end to end propagation delay, r_{in} (indicated by R_i above) is the rate of input to the decoder buffer, r_{out} (indicated by E_i above) is the output rate from this buffer, B_{start} is the decoder buffer occupancy prior to the retransmission period, B_{end} is its occupancy at the end of this period, and $T_{\max}$ is the maximum frame transmission time. Then, assuming that $2t_{prop}$ goes idle during ACK/NAK[12], we get

$$B_{end} = B_{start} + W \times T_{\max} \times r_{in} - (W \times T_{\max} + 2t_{prop} + B \times T_{\max}) \times r_{out} \tag{5}$$

We are mainly concerned with the extent of buffer underflow due to retransmissions. Therefore, to a first approximation, we neglect variations in r_{in} and frame transmission times(E_i). The theory is easily extendible to incorporate these variations. In fact, these give us further ways to prevent underflow by changing the source coding (changing E_i) or renegotiating with the channel (changing r_{in}; see [19,20] for details).

Therefore, assuming that the buffer was oscillating around its nominal value prior to the retransmission period, the extent of buffer underflow due to retransmissions is

$$U = (W \times T_{\max} + 2t_{prop} + B \times T_{\max}) \times r_{out} - W \times T_{\max} \times r_{in}, \tag{6}$$

with

$$0 \leq B \leq W \tag{7}$$

Again, to a first approximation, and for time scales in operation, we assume $r_{out} = r_{in}$

Hence,

$$U = (2t_{prop} + B \times T_{\max}) \times r_{out} \tag{8}$$

[12] For simplicity, we do not consider continuous ARQ protocols. Further, we assume that the first erred frame of a maximum of W buffered up is passed to the decoder only after its retransmission is received. Later packets have to be buffered till then to avoid reordering problems.

To accomodate this underflow, we propose to slow down the rate of display [13] . If we choke the output rate r_{out} by an amount Δr, then the buffer will return to its nominal value after a time

$$T = \frac{(2t_{prop} + B \times T_{\max}) \times r_{out}}{\Delta r} \tag{9}$$

For the scenario under consideration, $t_{prop} \ll B \times T_{\max}$, and $\Delta r / r_{out}$ can also be considered as fractional clock slow-down at the receiver. To simplify the display modification, we assume that slow down in display is possible only using frame skips (i.e., repeating frames in the display). Thus, the average number of display instants to be spread out or the total necessary shift in the display axis is,

$$N_s = T/D \tag{10}$$

where D is the average frame interval (e.g., 34 msec. for 30 Hz.). This shift is in addition to any frame-skips in the encoded stream itself (which is variable frame rate in this case). The above assumes that the network adaptation layer provides the necessary timestamp support for synchronization. Other dejittering mechanisms to absorb slight variations in network delay [7,13] (e.g., end-system traffic shaping or smoothing) can be applied in addition to the scheme mentioned here.

The elastic decoder buffer spreads the anticipated buffer underflow depending on the current frame content. In other words, the manner in which the N_s instants are spread out in time differs depending on content. The content which we have under consideration here is motion. If the current frame has high motion then the display axis is shifted slowly to avoid jerks, but if it has low motion, the shift is instantaneous. Notice that the receiver knows N_s after receiving W frames, and does not have to wait for the retransmitted B to start shifting the display axis.

As an illustration, we give below sample code for a single window size i.e., $W = 1$ and at 30 Hz. Important observations on the code are also mentioned.

Pseudo C code.

Step 0. $B^d_{init} = first\_frame\_size + \bar{m} + (\sigma/2)$ where $\bar{m}$ is the mean frame size and σ is the standard deviation in frame size.

Case (a). *if* $(display_i - arrival_i) < Threshold$ *or* (low_motion) *then* $display_i = (display_i)_{orig} + (N_s \times 34)$

Case (b). *if* $(display_i - arrival_i) > Threshold$ *and* $(high\_motion)$ *then* $display_{i+k-1} = (display_{i+k-1})_{orig} + \max(\left\lceil \frac{N_s}{2^k} \right\rceil, 1) \times 34$ for k=1,2,3 .... $K_{\max}$ where $K_{\max}$ is the minimum number satisfying $\sum_{k=1}^{K_{\max}} \max(\left\lceil \frac{N_s}{2^k} \right\rceil, 1) > N_s$

[13] This has benefits in terms of latency over re-negotiating R_i or changing E_i since MAC layer delays are quite high for the current system. However, video quality may suffer from jerks. Our objective is to minimize the perception of these jerks

Observations on Step 0.

This total startup buffering is useful if we shift the display axis slowly as in Case(b). In other words, $\bar{m} + (\sigma/2)$ is a cushioning term for slow shift in Case (b).

Observations on Case(a).

The shift in the display axis is instantaneous. This is because we can tolerate jerks with low motion. The first condition is used as an emergency measure and $Threshold$ is set at $\bar{m} + (\sigma/2)$. A low_motion high α frame is retransmitted according to Table 5.

Observations on Case(b).

It takes logarithmic time to provide the necessary shift to the display axis, or $\mathrm{K_{max}} \leq \log_2 N_s$. In this case, we do not wish to accomodate the entire shift on one frame since the frame has high motion and we wish to avoid jerks. The amount of shift decreases exponentially and is most at instant i (equal to half of the total required). This is adopted since errors normally occur in bursts. By doing this, the effect of tails on following frames falls off fast. Therefore, we do not get big carry over terms from previous frames in case of burst. Further, the effect of these terms at a given frame is bounded from above by slippage required by the maximum sized frame.

For example, if W=1, and we consider a frame of size $\bar{m} + (\sigma/2) = 829$ bytes, its transmission time is about 276 msec. Hence, $N_s = 6 + 276/34 \approx 9$ skips. Depending on the frame content, this shift would be given instantaneously (Case a) or over $\mathrm{K_{max}} \leq \log_2 N_s = 4$ to be displayed frames (Case b). Results of this algorithm are discussed in the following section.

5.4 ARQ Results

The ARQ scheme used is also known as Type-I ARQ [14]. The analysis was carried out assuming header + α based FEC. We used $t(\alpha_{high}) = 6, t(\alpha_{medium}) = 6, t(\alpha_{low}) = 5, t(header) = 9$ (as defined in Table 3). We use different degrees of interleaving to vary the effective burst lengths. Figure 7 shows the error distribution after FEC correction, but without any retransmissions. It can be seen that even if we make binary decisions at a frame level, errors normally occur in bursts.

Figure 8 shows the number of frame errors after retransmissions. It can be seen that a significant improvement can be obtained by using ARQ (compare with Figure 7) . Figure 8 assumes that late packets are useful. However, we can control the number of late frames by reducing the effective frame display rate at the receiver nominally. This was done using the algorithm mentioned above.

[14] In Type-I ARQ, FEC is applied to "all" frames and ARQ is used for erred frames after FEC decoding.

Figures 9 (a) and (c) shows the arrival and display deadlines without any shift to the display axis. The dashed line shows the target display deadline and the solid line is the arrival time. It can be seen that the display deadlines cannot be met if we use ARQ without any shift to the display axis. The error distribution is shown in (c) to illustrate how retransmissions cause display deadlines not to be satisfied. The time it takes for arrival times to start falling behind depends on the extent of interleaving and error protection used.

Figures 9 (b) and (d) shows the arrival and display deadlines with shifts to the display axis according to the algorithm above. The arrival curve is maintained below the display curve in the long run. By giving an elastic content-dependent shift to the display axis we are able to meet display deadlines while maintaining good visual quality. An equivalent representation to validate the above algorithm could show the receiver buffer occupancy.

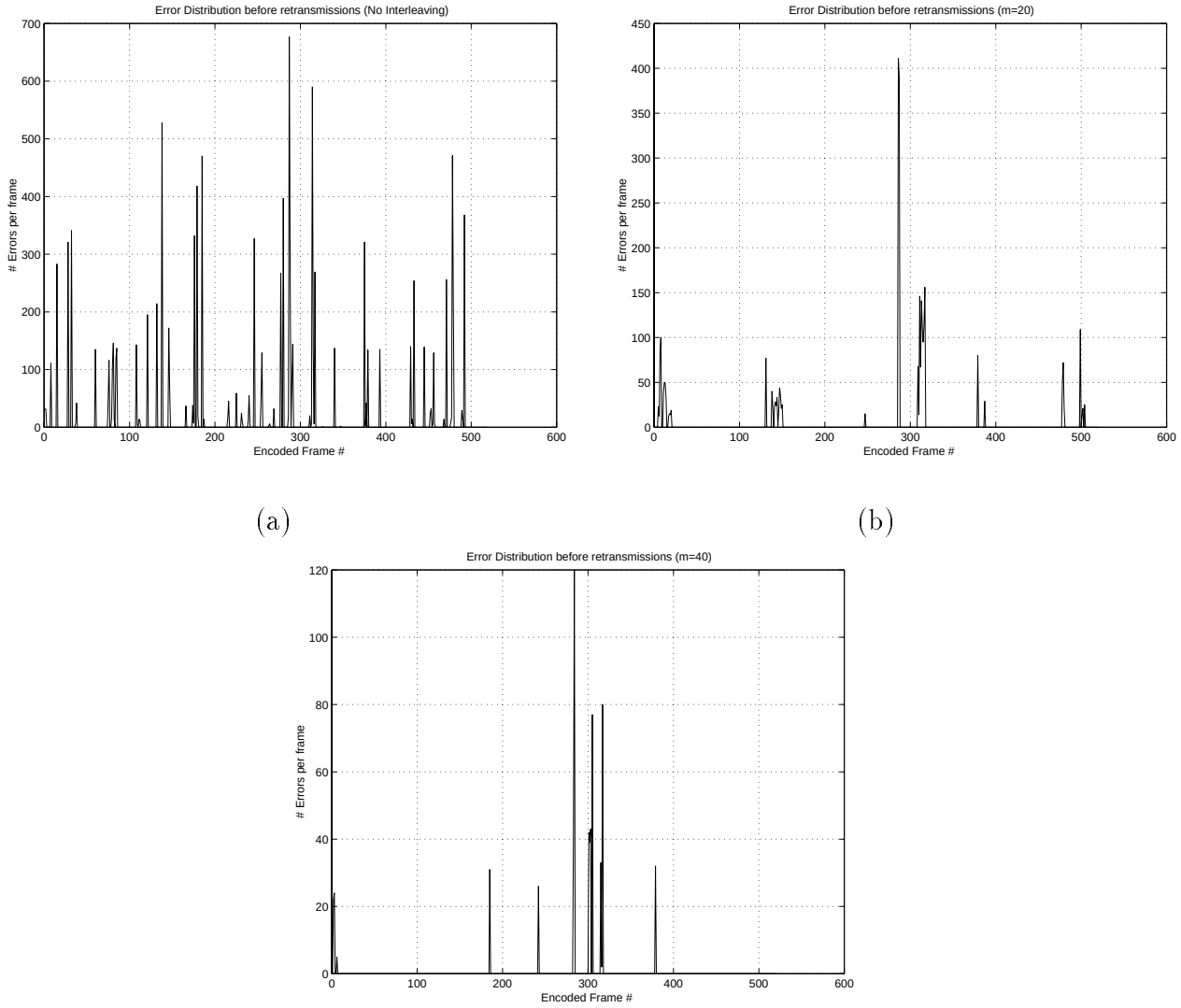


Fig. 7. *Error distribution before retransmissions. Channel BER (without any FEC)* $= 5.23\times 10^{-3}$. *Errors shown are in the encoded video stream. (a) No interleaving (b) Interleaving degree = 20, (c) Interleaving degree = 40*

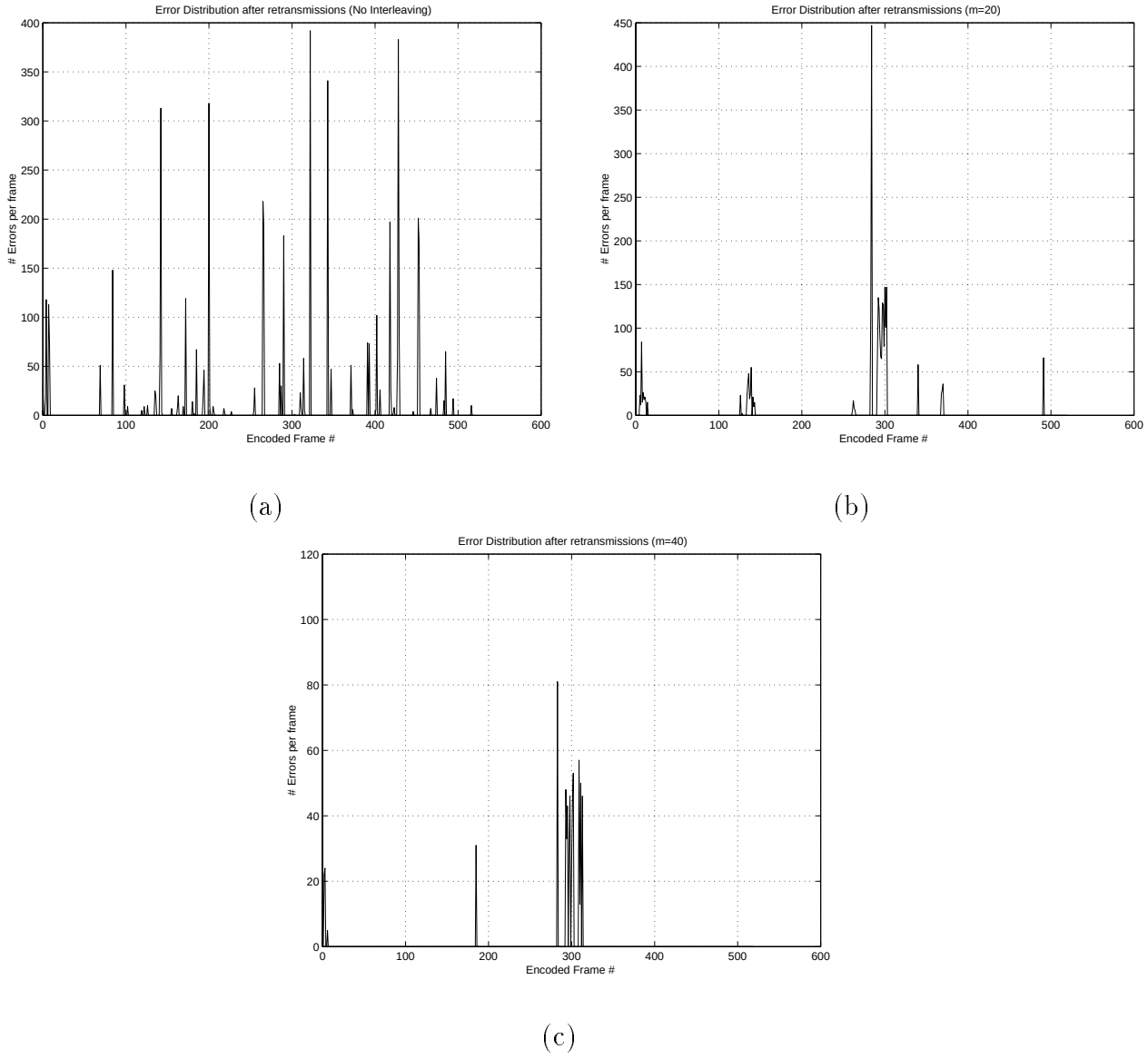


Fig. 8. *The number of frame errors after retransmissions. Errors shown are in encoded video. Channel BER (without any FEC) = 5.1 $\times$ 10^{-3}. (a) No interleaving, (b) Interleaving degree = 20, (c) Interleaving degree = 40,*

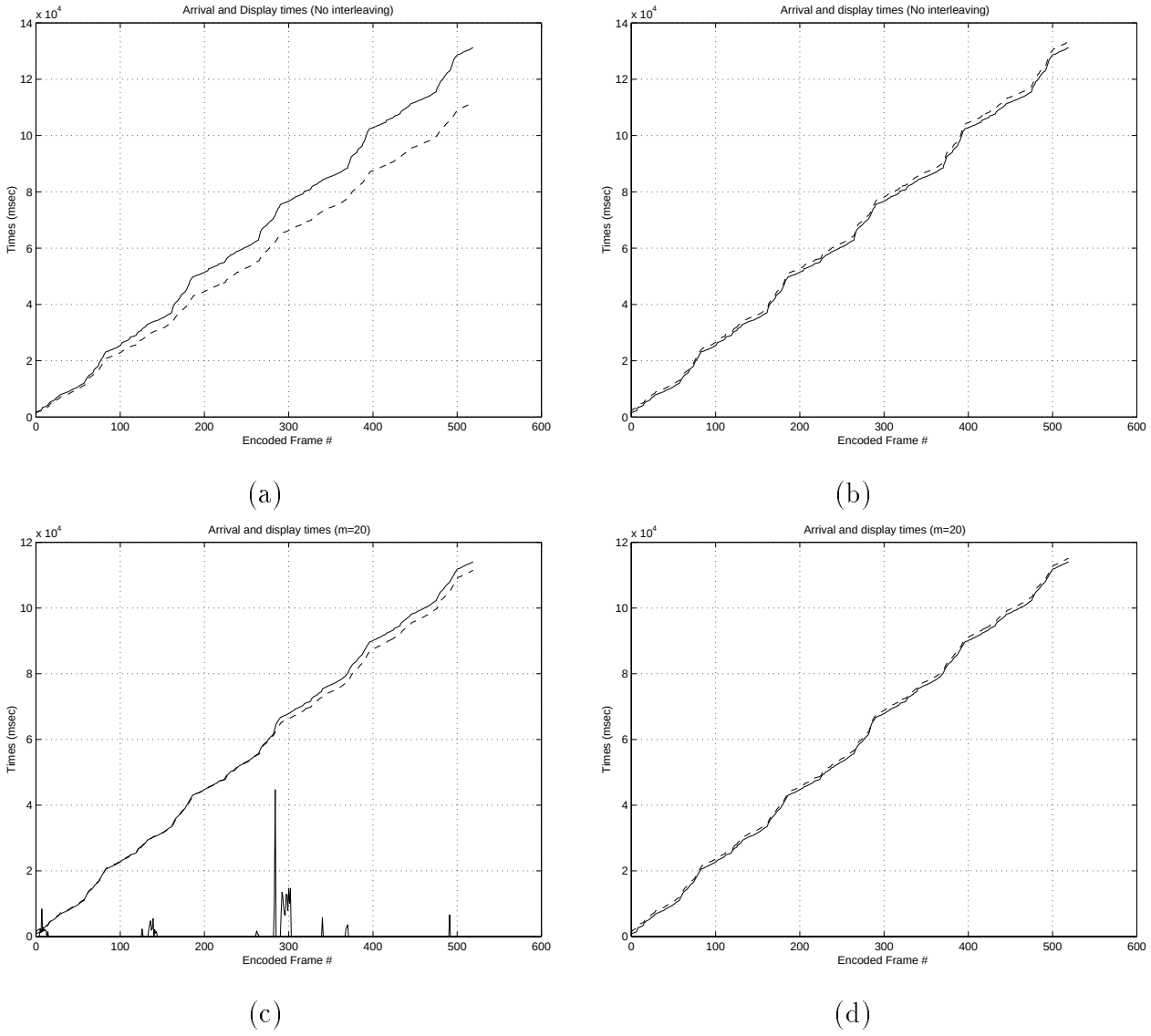


Fig. 9. *Arrival and display deadlines. Window size = 1. Dashed-Display, Solid - Arrival times. (a,c) without rate adaptation. (b,d) with rate adaptation (a,b) No interleaving (c,d) Interleaving degree = 20*

6 CONTRIBUTIONS AND SUMMARY

In this work, we presented a new approach for adaptive transmission of video over wireless channels. We propose a scheme for segmenting video into classes of various perceptual importance. We present a detailed mechanism for mapping between segmentation and resource allocation. The proposed approach can be used as an enhancement to the currently available scalability profiles, both at a frame and future video objects (as defined in MPEG-4). In particular, the extent of intra coding, scene changes, and motion based procedures are used to provide finer level of control for data segmentation. We present a comprehensive analysis on the incremental effect of these attributes using a practical mobile and wireless channel simulator. Based on our FEC results, we see that header and high intra (or scene change) based segmentation gives the largest increase in PSNR. The extent of intra coding (Partitions 2 and 3) and motion information can be used to further increase the perceptual video quality. In the second half of the paper, we present algorithms using selective repeat ARQ for one-way video applications. To accommodate the variable delay caused by the selective use of ARQ schemes, our approach also includes an "elastic" buffer before the decoder/display module to "absorb" the delay jitter. The buffer will absorb the rate variation caused by selective ARQs and provide a compatible interface to the decoder. We present efficient algorithms for buffer control in this case. The proposed framework using video content provides great synergy to future video coding standards, e.g., MPEG-4, using object-based video representations. Based on our results we argue that introduction of video content into video segmentation allows us to define various classes of importance which give us finer level of control. We believe that by using such techniques we can get better visual quality under given resource constraints.

7 FUTURE WORK

Currently, we are extending the approach to use information from late frames to refine video quality and prevent error propagation effects. Object based techniques for segmentation like those being proposed in the current MPEG-4 standardization efforts are being investigated. Also redundancies due to automatic object level motion tracking, content-scalabilities, and VOP based transport are being studied. We intend to apply similar approaches to Internet packet based video transport as well.

Acknowledgements

We would like to acknowledge Dr. Li-Fung Chang at Bellcore, Dr. M.T. Sun at University of Washington and Dr. A. Wong at Lucent Technologies for their help with the wireless simulator. We are also thankful to Dr. Amy Reibman at AT&T for her suggestions. Discussions with Dr. L.C. Yun (formerly at UC Berkeley) were also helpful. This work was supported in part by the industrial sponsors of the ADVENT project of Columbia University.

References

[1] E. Amir, S. McCanne, and M. Vetterli. A layered DCT coder for internet video. *Proceedings IEEE International Conference on Image Processing*, 1:13–16, September 1996.

[2] R. Aravind, R. Civanlar, and A. R. Reibman. Packet loss resilience of MPEG-2 scalable video coding algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(5):426–435, October 1996.

[3] E. Ayanoglu, P. Pancha, A. Reibman, and S. Talwar. Forward error control for MPEG-2 video transport in a wireless LAN. *ACM/Baltzer Mobile Networks and Applications Journal*, 1:235–244, December 1996.

[4] Belzer, Liao, and J.D. Villasenor. Adaptive video coding for mobile wireless networks. *Proceedings IEEE International Conference on Image Processing*, October 1994.

[5] E.R. Berklekamp, R.E. Peile, and S.P. Pope. The application of error control to communications. *IEEE Communications Magazine*, 25(4):44–57, April 1987.

[6] R. Clarke. *Digital Compression of Still Images and Video.* Academic Press, London, England, 1995.

[7] R.L. Cruz. A calculus of network delay, Part 1: Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[8] M.W. Garrett and M. Vetterli. Joint source/channel coding of statistically multiplexed real-time services on packet networks. *IEEE/ACM Transactions on Networking*, 1(1):71–80, February 1993.

[9] M Ghanbari. Two-layer coding of video signals for VBR networks. *IEEE Journal on Selected Areas in Communications*, 7(5):771–781, June 1989.

[10] M. Ghanbari. Postprocessing of late cells for packet video. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(6):669–678, December 1996.

[11] M. Ghanbari and V. Seferidis. Cell-loss concealment in ATM video codecs. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(3):238–247, June 1993.

[12] H. Gharavi and M. H. Partovi. Multilevel video coding and distribution architectures for emerging broadband digital networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(5):459–469, October 1996.

[13] ISO/IEC 13818-1 MPEG-2 H.222.0. Generic coding of moving pictures and associated audio : Systems, November 1994.

[14] ISO/IEC 13818-2 MPEG-2 H.262. Video, November 1994.

[15] ITU-T Draft H.263. Line transmission of non-telephone signals, July 1995.

[16] R.H. Han and D.G. Messerschmitt. Asymptotically reliable transport of multimedia/graphics over wireless channels. *Proceedings Multimedia Computing and Networking*, pages 29–31, January 1996.

[17] L. Hanzo and J. Streit. Adaptive low rate wireless videophone schemes. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(4):305–318, August 1995.

[18] T. R. Hsing, L.-F. Chang, A Wong, M.-T. Sun, and T.-C. Chen. A real-time software based end-to-end wireless visual communications simulation platform. *Polytechnic University Symposium on Multimedia Communications and Video Coding*, November 1995.

[19] C.-Y. Hsu, A. Ortega, and M. Khansari. Rate control for robust video transmission over wireless channels. *Proceedings, Visual Communications and Image Processing (VCIP)*, February 1997.

[20] C.-Y. Hsu, A. Ortega, and A.R. Reibman. Joint selection of source and channel rate for VBR video transmission under ATM policing constraints. *IEEE Journal on Selected Areas in Communications*, 15(6):1016–1028, August 1997.

[21] W.C. Jakes Jr. *Microwave Mobile Communications*. John Wiley and Sons, 1994.

[22] G. Karlsson and M. Vetterli. Packet video and its integration into the network architecture. *IEEE Journal on Selected Areas in Communications*, 7(5):739–751, June 1989.

[23] M. Khansari, A. Jalali, E Dubois, and P. Mermelstein. Low bit-rate video transmission over fading channels for wireless microcellular systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(1):1–11, February 1996.

[24] W-M Lam and A.R. Reibman. An error concealment algorithm for images subject to channel errors. *IEEE Transactions on Image Processing*, 4(5):533–542, May 1995.

[25] B. Lamparter and M. Kalfane. The implementation of PET. *TR-95-047, International Computer Science Institute*, August 1995.

[26] H. Liu and M. E. Zarki. Data partitioning and unbalanced error protection for H.263 video transmission over wireless channels. *Polytechnic University Symposium of Multimedia Communications and Video Coding*, November 1995.

[27] J. Meng and S.-F. Chang. CVEPS: A compressed video editing and parsing system. *ACM Multimedia Conference*, 2419, November 1996.

[28] J.W. Modestino, D.G. Daut, and A.L. Vickers. Combined source-channel coding of images using the block cosine transform. *IEEE Transactions on Communications*, COM-29(9), September 1981.

[29] ISO/IEC JTC1/SC29/WG11 N1483 MPEG96. Systems working draft version 2.0, November 1996.

[30] ISO/IEC JTC1/SC29/WG11 N1484 MPEG96. Systems verification model version 2.0, November 1996.

[31] ISO/IEC JTC1/SC29/WG11 N1808 MPEG97. Description of core experiments on error resilience aspects in MPEG-4 video, 1997.

[32] A.R. Noerpel, Y.-B. Lin, and H Sherry. PACS: Personal access communications system - a tutorial. *IEEE Personal Communications*, 3(3):32–43, November 1996.

[33] V. Parthasarathy, J.W. Modestino, and K.S. Vastola. Design of a transport coding scheme for high-quality video over ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):358–376, April 1997.

[34] A. Puri, A.R. Reibman, R.L. Schmidt, and B.G. Haskell. Considerations in ISO MPEG-4 and ITU-T mobile video standards. *Proceedings 3rd. International Workshop on Mobile Multimedia Communications (MoMuC-3)*, September 1996.

[35] K. Ramachandran, A. Ortega, K.M. Uz, and M. Vetterli. Multiresolution broadcast for digital HDTV using Joint Source/Channel Coding. *IEEE Journal on Selected Areas in Communications*, 11(1):6–23, January 1993.

[36] J.M. Reason, L.C. Yun, A.Y. Lao, and D.G. Messerschmitt. Asynchronous video: Coordinated video coding and tranport for heterogeneous networks with wireless access, Mobile Computing, H.F. Korth and T. Imielinski Ed., Kluwer Academic Publishers, Boston, MA, 1995.

[37] Amy R. Reibman and B Haskell. Constraints on variable bit rate video for ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2(4):361–372, December 1992.

[38] R. Stedman, H. Gharavi, L. Hanzo, and R. Steele. Transmission of sub-band coded images via mobile channels. *IEEE Transactions on Circuits and Systems for Video Technology*, 3:15–26, February 1993.

[39] J. Streit and L. Hanzo. Quadtree-based reconfigurable cordless videophone systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(2):225–237, April 1996.

[40] R. Talluri etal. A robust, scalable, object-based video compression technique for very low bit-rate coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):221–233, February 1997.

[41] J. Zhang, M.R. Frater, J.F. Arnold, and T.M. Percival. MPEG-2 video services for Wireless ATM Networks. *IEEE Journal on Selected Areas in Communications*, 15(1):119–128, January 1997.