Traffic Shaping for Variable-Bit-Rate MPEG-2 Video

Il-Sung Lee

Department of Electrical Engineering
McGill University
Montréal, Canada

August 1997

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

0-612-37266-9

Abstract

As digital video services become more prominent, the strain that they place upon networks will become more pronounced. The network of choice for carrying these transmissions will be the Broadband Integrated Services Digital Network using the Asynchronous Transfer Mode (ATM) transmission protocol. In particular, ATM is well suited for handling variable-bit-rate (VBR) transmissions because it is capable of statistical mulitplexing. VBR coding is also advantageous because it is conducive for producing consistent image quality. This thesis outlines the conversion of a software MPEG-2 video coder to operate as a VBR coder and then as a traffic shaping one. The rate controller for the traffic shaping coder is based on the Leaky Bucket policing algorithm to ensure conformity to the network traffic descriptors.

First, comparisons are made between constant-bit-rate encoded sequences and their VBR counterparts. As expected, the VBR sequences possess superior average peak signal to noise ratios (PSNRs) while maintaining a more consistent PSNR between frames. Secondly, traffic shaped sequences are introduced for comparison. Compared to the VBR case, these sequences exhibit a less erratic bit rate profile but also a lower average PSNR. However unlike the open-loop encoded data, the traffic shaped bit streams are not likely to encounter any network intervention when transmitted in any realistic network scenario. The effects of varying each network traffic descriptor (peak bit rate, sustained bit rate, and maximum burst size) are examined and the projected network-induced bit losses in the absence of traffic shaping are provided.

Sommaire

À mesure que les applications vidéo numériques prennent une place de plus en plus importante, le fardeau qu'elles mettent sur les réseaux de communication se fait de plus en plus lourd. Le réseau de choix pour acheminer ces données sera le Réseau Numérique à Intégration de Services (ISDN) à large bande utilisant le protocole de transmission à mode de transfert asynchrone (ATM). En particulier, le ATM est bien adapté pour les transmissions à débit de données variable (VBR) puisqu'il permet le multiplexage statistique. Le codage VBR est aussi avantageux puisqu'il permet une qualité d' images qui est plus constante. Cette thèse traite de la modification d'un codeur vidéo MPEG-2 pour agir comme codeur VBR et ensuite comme un codeur permettant de façonner le trafic. Le contrôle du trafic de ce dernier est basé sur la politique du seau percé ("leaky bucket") afin de se conformer aux descripteurs du trafic du réseau.

Tout d'abord, des comparaisons sont faites entre des séquences vidéo codées à débit constant et à débit variable (VBR). Tel que prévu, les séquences codées VBR présentent un rapport signal à bruit (PSNR) moyen supérieur tout en conservant un PSNR plus consistant d'une trame à l'autre. Ensuite, des séquences façonnées pour le trafic sont présentées pour fins de comparaison. Comparées au cas VBR, ces séquences présentent un débit moins irrégulier mais aussi un PSNR moyen plus faible. Cependant, contrairement aux données codées en boucle ouverte, les flux de bits ("bit streams") façonnés pour le trafic n'auront probablement pas à subir une intervention du réseau lorsque transmis dans un scénario de réseau réaliste. Les effets de varier chaque descripteur du trafic de réseau ("peak bit rate", "sustained bit rate", "maxiumum burst size") sont examinés et les pertes en terme de bits, prévisibles sur le réseau sont aussi fournies.

Acknowledgments

First and foremost, I would like to thank Dr. Éric Dubois for extending an invitation to supervise this project and introducing me to field of digital video coding. The volume of knowledge and skills gained throughout this project is immense and for this I will always be indebted. I have sincerely appreciated his guidance and patience. I must also acknowledge my appreciation for the financial assistance he, along with the Canadian Institute for Telecommunications Research, has generously provided.

Next, I wish to thank Dr. Nadia Baaziz for having developed the original INRS MPEG-2 coder, as well as other various utility software, used throughout the course of this research. I am grateful for her assistance in the early stages of my work and the patience she afforded me in discussing and explaining the unclear and, at times, puzzling concepts surrounding the software and the MPEG standard.

To INRS-Télécommunications, I offer my gratitude for providing truly excellent facilities and for lending me desk-space; a phone would have warranted my utmost gratitude.

To Dr. Maier Blostein, I thank him for his early guidance and willingness to be my supervisor at the University of McGill. Now he will no longer have to ask me "are you still here?"

I would like to express my appreciation to Dr. Stephane Coulombe and Sonia Aissa for their assistance in translating the abstract into French, a feat, which if I had undertaken, might have brought me considerable ridicule. Also at INRS, I would like to sincerely thank Anthony Mancini for his frequent programming support and his assistance in handling the affairs of my thesis submission during my absence from Montréal. It is up to you to continue the tradition.

I would like to thank my family for their unbelievable support and understanding during my academic life, especially during this latest chapter at McGill; it means more to me than I can express.

Finally, but certainly not least, my thanks go to the all good friends I have made at McGill and INRS. I would particularly like to mention a special group students with whom I have had the fortune to become roommates, and then good friends, when I first commenced my postgraduate studies. They have made my sojourn in Montréal enjoyable and memorable and I owe them a lot. To Raymond Lee, thank you for graciously allowing me to stay with you during the last stretch of writing this thesis and, more importantly, for being a true friend. To Emmy Choi, I will never forget your support, encouragement, and above all, friendship during this last year. All will truly be missed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of digital video has gained increasing prominence in recent times. Because of the marketability and mass appeal to consumers, research into digital video has become very prevalent. A digital representation of a video sequence is unquestionably superior to a conventional analog representation because of the high stability and reliability during storage or transmission and also because of the increased flexibility afforded by digital signal processing. The primary example of the latter is the important issue of digital video compression. The most recognized and accepted method is the *MPEG-2* algorithm developed by the ISO/IEC Moving Pictures Expert Group. With MPEG-2, video sequences can be compressed to an astonishing five percent or less of their original size and still manage to retain an excellent level of image quality. However, this all comes at the expense of a considerable amount of computational complexity. Fortunately, VLSI technology is rapidly making this less of a concern. Dedicated chips that are capable of encoding or decoding MPEG-2 bit streams in real time or close to real time are readily available. In addition, the performance of personal computers is increasing at such a rate that specialized decoding chips may soon become unnecessary. For instance, Intel has recently released a new class of processors called the Pentium MMX and the 200MHz version has the computational power to decode and display a full screen MPEG-2 bit stream at 20 frames per second.

Terms such as HDTV (High Definition Television), Video on Demand, DVD (Digital Video Disc), Video CD, and DBS (Direct Broadcast Satellite) now permeate the media. All deal with digital video and at least HDTV, DVD, and DBS deal exclusively with the MPEG-2 algorithm. Finalized in 1994, MPEG-2, formally known as *ISO/IEC 13818-2: Generic Coding of Moving Pictures and Associated Audio Information*, is a lossy compression algorithm offering the flexibility to handle sequences ranging from SIF to HDTV format. Essentially, MPEG-2 employs a block-based DCT compression for intra-frame coding and block-based motion compensation for inter-frame coding. The Institut national de la recherche scientifique (INRS) Télécommunications possess a C language software implementation of the MPEG-2 algorithm and follows the method demonstrated by the MPEG-2 Test Model 5.

The ability to transmit the encoded digital signals is a key motivation, both technically and economically, for the research efforts spent on digital video. Being able to receive video information from remote locations (e.g., teleconferencing and Video on Demand) is particularly alluring as this affords the user greater flexibility and convenience. In addition, consumer cable television transmission will inevitably switch to a digital system in the near future, especially as HDTV becomes available and gains acceptance. From the point of view of the service provider, a great incentive to switch to digital formats is that new techniques have been developed (vestigial sideband, quadrature amplitude modulation) that enable a single 6MHz analog TV channel to carry a single digital HDTV signal or four 6Mbps MPEG-2 programs. Unfortunately, the transmission of digital video presents some difficulties. Compared to other digital applications, video bit streams tend to demand a great amount of transmission bandwidth, meaning that high speed networks must be a consideration. One such network ideally suited for this task is the Broadband Integrated Services Digital Network (B-ISDN) which is envisioned as the network of the future. The B-ISDN was created with the knowledge that visual communication services are predicted to be the primary services in the future. Flexibility was a key concern in the design of the B-ISDN and for this reason, the *Asynchronous Transfer Mode*, or ATM for short, was devised as the transport protocol to manage these networks. ATM offers the B-ISDN, or any other network, great flexibility in that it can easily assimilate a wide variety of different types of traffic and, in particular, variable-bit-rate (VBR) traffic. Furthermore, with VBR sources, ATM has the ability to statistically multiplex signals together which raises the efficiency of bandwidth utilization and effectively increases the number of streams that a network can carry beyond what its bandwidth would indicate. Luckily, the coding and transmission of video signals in variable bit rate is usually preferable to using constant bit rate (CBR). Unlike the characteristic time-varying image quality of CBR coding, an MPEG-2 video sequence encoded with a variable rate is capable of preserving a very consistent quality level for the duration of the sequence. Of course, the disadvantage of transmitting a VBR signal is the difficulties placed upon the network in managing such traffic.

The obvious problem for a network that allows its connections to vary in bit rate over time is how to cope with the possibility of channel overload. In other words, how does the network prevent the sum of the channel bandwidths occupied by the individual connections from exceeding the physical network capacity at any given time? A network overload incited by one stream would be disastrous in that any other stream concurrently using the same link may suffer some loss as a side effect. The solution is to ensure that each traffic source is "well-behaved." The first step is to obtain a description of the source so that the time-varying rate behavior can be predicted. One common method to describe a source is through a set of statistical descriptive parameters. In the context of ATM, three traffic descriptors are used and they are named the *peak rate* (R_P), *sustained rate* (R_S), and the *maximum burst size* (mbs). With these parameters, the network can monitor each source and take corrective measures if any traffic descriptor for that source is exceeded in order to maintain the network's integrity and uphold the quality of service (QoS) promised to the other sources. The network function that detects nonconforming sources and takes appropriate actions is called the network *policer.* In particular, to determine the acceptability of a source, an algorithm called the *Leaky Bucket*, or the *Generic Cell Rate Algorithm* (GCRA) as labeled by the ATM Forum, is employed within the policer. In general, the policer handles nonconforming sources by discarding packets of information belonging to that source until the traffic descriptors are satisfied. Obviously, the loss of any information can be harmful to a transmission, but especially for MPEG-2 video since an error may propagate to successive video frames and there is usually no opportunity for re-transmissions. For this reason, a source is often *traffic shaped* to ensure that all of its transmitted data will reach the client with high probability.

Many researchers have investigated the issue of traffic shaping bit streams, and in particular, MPEG-2 bits streams. All use a rate controller to monitor the fullness of a rate buffer and then precipitate a reactive adjustment to the coder using some quantizer-control algorithm. Whenever rate control of video is discussed in the literature, the scenario usually either involves real-time compression and transmission or simply real-time traffic shaping of a previously coded sequence. In either case, a shortcoming exists in that whenever a traffic violation occurs, only a reactive measure that will effect future data can take place. The violating segment of data has already entered the network. In addition, the buffer sizes must necessarily be large to accomplish the traffic shaping. In this thesis, a method designed specifically for the traffic shaping of off-line encoded sequences is explored. Furthermore, rather than simply exercising preventative steps effecting only the future output, the traffic shaping coder described here is capable of taking retroactive preventative steps to ensure that all segments of the bit stream, even the current one, conforms to the limitations outlined by the network traffic descriptors. The coder incorporates the ATM Forum's specification of the Leaky Bucket algorithm in its rate control to ensure that the judgment of the rate controller mimics that of the actual network.

The layout of this thesis is as follows. The next chapter provides background material on the concepts addressed in this thesis and is broken into three sections – the MPEG-2 Algorithm, Constant and Variable Bit Rate Coding, and ATM. Chapter 3 is intended to present a validation of the superiority of variable-bit-rate coding over constant-bit-rate coding. The bit-rate profiles and luminance peak signal to noise ratios (PSNR) are discussed to highlight the differences. The VBR coder is obtained by modifying the original INRS MPEG-2 coder and certain modifications are discussed. To acquire the constant-bit-rate data, results are generated using the rate-constrained coder (described in Chapter 4) as well as a coder utilizing the well-established method of the MPEG-2 Test Model 5. The additional flexibility offered by the rate-constrained coder allows the bit rate to be held constant over a much finer interval than with the Test Model 5. Chapter 4 investigates the issues surrounding the development of the rate-constrained coder and makes observations derived from its operation. A detailed description of its components and the imposing technical challenges is given. In addition, experiments that provide insight into the effects of varying each of the three traffic descriptors individually are outlined. In particular, the impact upon the bit rate and, more importantly, the qualitative and quantitative picture quality is assessed. Also investigated is the projected network bit loss that can be expected from the absence of traffic shaping. Finally, Chapter 5 presents a summary of the contributions of this thesis and provides suggestions for future work. For convenience, an appendix providing a quick definition of some of the main terms and variables used throughout this thesis is included to facilitate the understanding of the work. The last two appendices contain the graphical and numerical chrominance information resulting from the experiments conducted in Chapters 3 and 4. This material is included for the most curious of readers who feel that knowledge of the chrominance behavior may provide a more complete result.

Chapter 2

Background

Digital video sequences, in raw form, are simply too large to be economical to transmit or store. Fortunately, efficient algorithms, such as MPEG-2, exist that can compress the video information into very manageable sizes. Moreover, the technology is in place to decompress and display these sequences in real time. The transmission of digital video poses another problem as, even in their compressed form, video sequences are still the greatest consumers of bandwidth. However, the networks of the future, such as the Broadband Integrated Services Digital Network, or B-ISDN, will have the capability to easily handle video bit streams. The *Asynchronous Transfer Mode* (ATM) is the most prominent candidate to be the protocol for these high bandwidth networks. ATM, which is a packet-based switching method, is very capable of handling large transmissions, even if the transmission bit rates vary with time, which is common with digital video. To prevent any traffic from overloading the entire network, ATM employs network monitoring functions to police the network. This situation of transmitting packetized video signals can succinctly be referred to as *packet video* [1].

The following sections introduce the general functionality of the MPEG-2 algorithm as well as the ATM protocol with mention of the network policing function. Also, the motivations behind constant and variable-bit-rate encoding, which are important to both MPEG-2 and ATM, are discussed.

2.1 The MPEG-2 Video Encoding Standard

Until recently, nearly all visual media have been stored and transmitted as analog signals. However, the rapid advancements in VLSI technology are quickly facilitating the replacement of analog video with digital video. Visual signals stored digitally are extremely stable and can be replayed without showing signs of degradation. In addition, a great amount of flexibility is introduced in manipulating the video signal through the possibility of digital signal processing.

Digital video requires an impressive amount of bits to accurately represent a video sequence of any appreciable length. This is understandable when one considers that video is a collection of time varying images spaced at regular intervals through time. For instance, a sequence spanning just one second of video images in ITU-601 format can occupy in the neighborhood of twenty megabytes (Mb) of computer storage [2]. For transmission, this translates to a 160 megabits per second (Mbps) channel. This represents a considerable strain on the storage or transmission medium, especially if several long sequences are concurrently involved.

In order to make digital video transfers to the public economically and technologically feasible, source coding (compression) becomes a necessity. By far, the most recognized and widely supported algorithm to achieve video compression is the *MPEG* standard. However, the second phase of the standard, *MPEG-2*, is quickly replacing its predecessor. Using the MPEG-2 algorithm, a video sequence can be compressed to about five percent of its original size and still retain enough information to allow a very good playback quality. To achieve this level of compression, a lossy algorithm is used to "intelligently" discard information so that the effects on the perceptible image quality is kept to a minimum. The continuation of this section provides a general overview of the standard. A plethora of publications exist which provide a more comprehensive discussion of the MPEG-2 standard; these include Puri [3], Le Gall [4], and Tudor [5]. Other references, such as Dubois [6] and Netravali and Haskell [7] discuss video coding in a more general context but is easily applicable to MPEG-2.

2.1.1 Overview

Research into rate compression algorithms for digital image sequences has been ongoing for nearly two decades. However, a strong interest in processing digital video and designing coding algorithms only began in 1982 when the CCIR Recommendation 601 (now ITU-R 601) defined the encoding formats for the production of digital television [8]. Since then, the coding algorithms for digital video have been standardized; first, by the CCIR (now the ITU) with the H.261 standard in 1990 and then by the ISO/IEC with the introduction of the MPEG standard in 1991 [9].

MPEG-2, formally known as *ISO/IEC JTC 1/SC 29/WG 11, 13818: Generic coding of moving pictures and associated audio information*, was officially standardized by the ISO/IEC Motion Picture Experts Group in 1994 [10]. This standard was meant to be a broader extension of the original MPEG-1 standard. Unlike the first standard, which was designed to be used to code progressively scanned video for CD storage, and consequently optimized for bit rates at around 1.5 Mbps, MPEG-2 is directed at broadcast formats and encompasses a much wider range of operating bit rates (1.5 to 20 Mbps). This includes the capabilities to efficiently encode interlaced video and HDTV sequences. MPEG-2 essentially inherited all the capabilities of the predecessor, however, there are quite a few notable differences. Some of second standard's new features include the ability to handle interlaced sequences, half-pixel motion vector accuracy, user-selectable DC precision, and four different scalable modes [10].

Essentially, the MPEG video compression algorithm relies upon two basic techniques to achieve compression. It uses transform-domain-based compression to reduce spatial redundancy and block-based motion compensation to reduce the temporal redundancies. There are three distinct types of frames employed by MPEG corresponding to three different methods of compression. These frames are labeled, *Intra* or I frames, *Predictive* or P frames, and *Bi-directional* or B frames.

2.1.2 Intra-frame coding

The first encoded frame of a MPEG-2 sequence is always an I frame. Because all their compression is attributed to spatial compression only, I frames are independent from all other frames. This has the benefit that I frames are immune to all errors that may have propagated from previous frames. On the other hand, without taking advantage of the temporal redundancies, I frames are not able to achieve nearly the compression levels of the other types of MPEG-2 frames and are found to require the largest number of bits to encode. It is because of this liability that they are usually spaced several frames apart. This distance, referred to as N in MPEG jargon, is typically 12 to 15 frames. This distance has been determined empirically to be a good compromise between compression level and image quality. Intra frames are used as references to the neighboring B and P frames and are also employed to facilitate the so called "trick modes", such as fast forward and fast reverse, specified in the MPEG standards [10].

In MPEG-2, spatial compression is performed on a block-by-block basis. Each I frame is segmented into eight-by-eight pixel blocks. The routine for compressing each block is outlined in figure 2.1. A discrete cosine transform (DCT) is the preliminary

Fig. 2.1 Compressing an I frame block

step performed on the block. For a typical block taken from a natural image, the transform tends to concentrate the energy in the lower frequency coefficients. This is a direct result of the spatial statistical redundancies present in the original image [7]. Next, the coefficients are *quantized*; by introducing a controlled amount of distortion, a high degree of bit rate reduction can be achieved. The higher frequency coefficients are typically near zero and consequently rounded to zero. The coefficients are then scanned using a *zigzag* pattern starting from the DC position as shown in figure 2.2. The ordering is done in this manner to increase the probability of having a long run of zeros which directly benefits the next step, the *run length coding*. Finally, run length codes are *entropy coded* (variable length coded) using a Huffman-like representation.

Quantization

The quantization stage is the sole source of the information loss. However, an effort is made to exploit the subjective redundancies present in all pictures. Specifically,

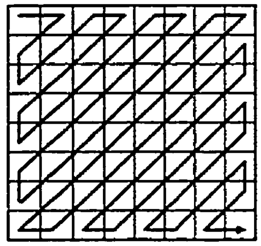


Fig. 2.2 DCT scan pattern

subjective redundancy refers to the parts of the visual information that are either imperceptible to the human eye or are too expensive to code [7]. Since noise and errors are less noticeable in the higher frequency regions of the image (for example, edges and detailed areas), the high frequency components of the DCT transform are more coarsely quantized than the low frequency ones. An important consideration is that the numerical precision of the coefficients can be adjusted by the encoder. A parameter, labeled Q, is used to control the quantization scaling factor. The value of Q is usually an integer between one and thirty-one; the actual quantization step is obtained by multiplying Q with the pre-stored quantization matrices for the luminance and the chrominance. If the quantization is done more coarsely, a larger portion of the high frequency DCT coefficients are likely to be rounded to zero resulting in a reduction in the number of bits required to encode the block. Obviously, this will have a direct impact upon the image quality, but as long as the adjustment is not too dramatic, the visual quality will still remain acceptable.

2.1.3 Inter-frame coding

Since the majority of information in a video sequence is found in the movement occurring within the scene, it should be no surprise that any video compression algorithm, including MPEG, uses motion compensation to achieve most of its compression. Essentially, motion compensation uses the premise that a point in the picture will closely follow a trajectory in the three-dimensional video space. The current picture is then constructed by repositioning the points from the reference picture according to their motion vectors (figure 2.3). MPEG uses block based motion estimation which assigns a motion vector to each 16×16 pixel block in a reference frame. P frames inter-

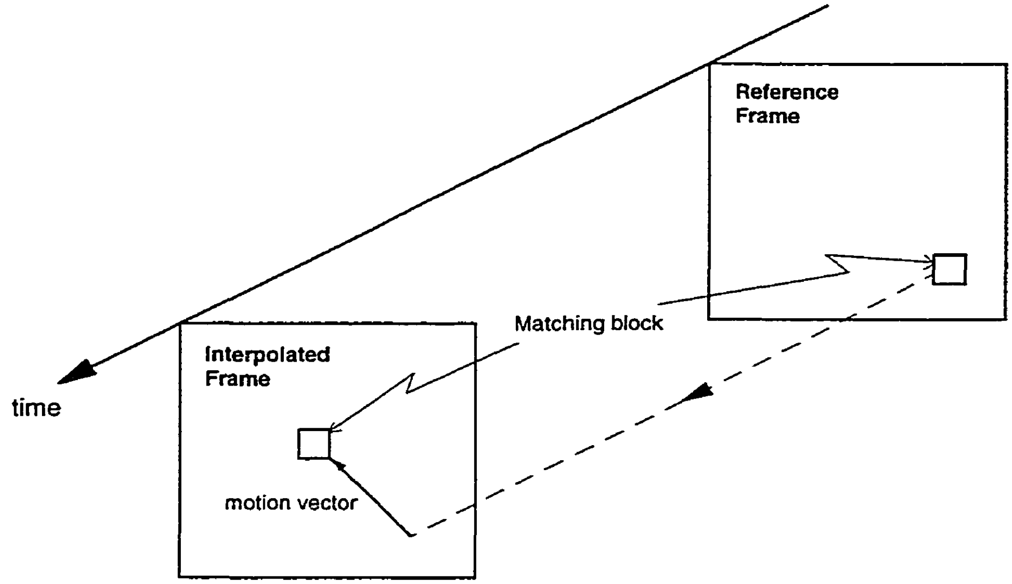


Fig. 2.3 Motion compensation

polate the image entirely from the most recent I or P frame. In addition, P frames serve as references for other motion-predicted pictures. The distance between consecutive predictive frames is given the descriptor M. B frames, on the other hand, can use a combination of two motion vectors. Their pictures are derived from the motion compensated images of the previous reference frame and the following reference frame. Thus, by taking advantage of the non-causal temporal information, further compression is achieved. This can immediately be seen for the case of object occlusion. However, the image may be interpolated using only one of the reference frames if this yields a better result. Additionally, the inter-frame pictures have the option of using zero prediction, which really means that the image will be intra coded. All these encoding choices can be made on a block-by-block basis. Figure 2.4 indicates the motion compensation frame dependencies for a sequence with $M = 3$ and $N = 15$. The numbers displayed in the lower part of each frame indicate the order in which they are encoded. The segment of the video that spans one I frame to the next I frame is called a *group of pictures* (GOP) in MPEG.

An MPEG-2 coder will process an I or P frame and then jump ahead M frames to the next reference frame before encoding the intermediate B frames. This is a direct consequence of the non-causal nature of the B frames. Because of this frame

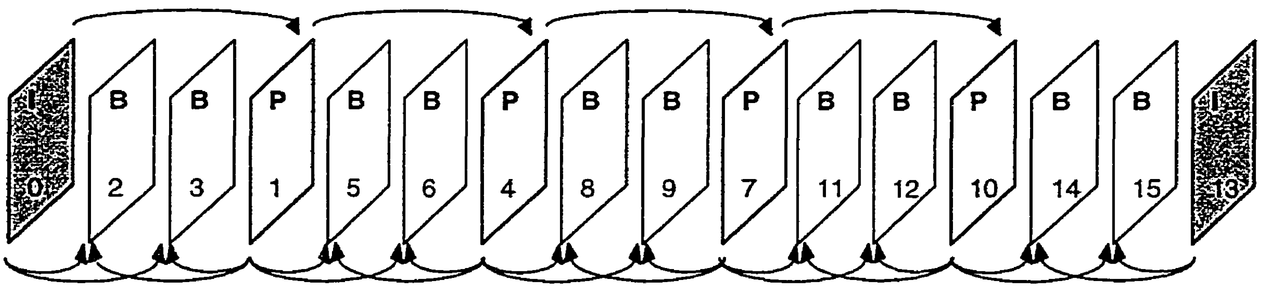


Fig. 2.4 MPEG Group of Pictures for $M = 3$ and $N = 15$

re-ordering during the encoding process, the first GOP produced by the coder, assuming $N = 15$, will have the frame composition ordered as IPBBPBBPBBPBBIB while every subsequent GOP will have the slightly different ordering BPBBPBBPBBPBBIB. Upon close examination of the two sequences the first sequence is found to have an extra I frame. This is an important observation when considering the transmission of the encoded GOPs because an extra I frame will noticeably increase the bit rate for the first series of frames compared to the bit rates for the following group of pictures. The reordering of the frames to their correct temporal positions becomes the responsibility of the decoder; this is carried out by using a system of frame buffers.

Motion compensation requires more than simply relaying the required motion vectors. While translational objects or background motion are well described by motion vectors, video seldom contains this type of movement alone. Other more complex scene changes such as zooming effects, lighting changes, and partial or entire object occlusion are very common. Also, there is the problem of describing objects entering or exiting the scene to consider. For this reason, the motion vectors are transmitted with the accompanying error signal. A local decoder in the encoder will construct the motion compensated image and then calculate and encode the prediction error. The error is encoded using the same procedure as the one used to encode the I frames. Figure 2.5 shows the general layout of the motion-compensated coder.

The calculation of the optimum motion vectors is by far the most time consuming aspect of the encoding process [11]. In terms of computational intensity, I frames are the least intensive while the B frames are the most. In terms of compression, the same is true. I frames achieve the least compression as they only take advantage of spatial redundancies while the B frames are capable of the most compression since they take

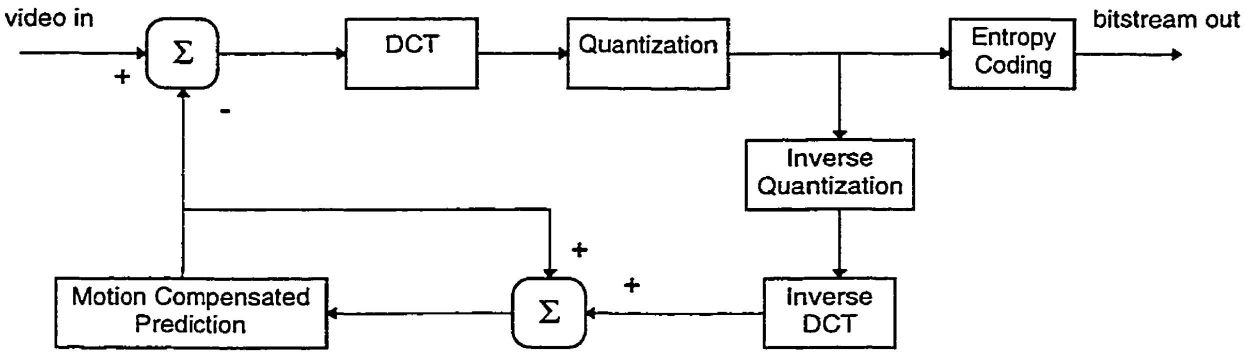


Fig. 2.5 Motion compensated coder

advantage of temporal redundancies, both causal and non-causal [12]. Typically in a sequence, I frames will be represented by about two times more bits than P frames and about three times more bits than B frames.

2.1.4 Profiles and levels

To be able to handle the broad range of applications, MPEG-2 has a set of algorithm tools at its disposal. These tools allow the codec to support such features as interlaced video and scalable coding. The tools themselves are divided into subsets called *profiles.* The profiles are accompanied by *levels* which are limits on the coding parameters such as the sample rates, frames dimensions, and coded bit rates. Tables 2.1 and 2.2 provides a summary of the different profiles and levels available to MPEG-2 [13, 3].

Profile	*Description*
Simple	I and P pictures only 4:2:0 sampling matrix 8, 9, or 10 bits DC precision
Main	I and P and B pictures 4:2:0 sampling matrix 8, 9, or 10 bits DC precision
SNR	SNR scalable coding
Spatial	Spatial scalable coding
High	4:2:2 and 4:4:4 sampling matrix 8, 9, 10, or 11 bits DC precision

Table 2.1 MPEG-2 Profiles

Level	*Description*
Low	SIF video rate 352 × 288 resolution at 30 frames/s Up to 4 Mbps
Main	ITU-R 601 video rate 720 × 576 resolution at 30 frames/s up to 15 Mbps
High-1440	1440 × 1152 resolution at 60 frames/s up to 60 Mbps
High	1920 × 1152 at 60 frames/s up to 80 Mbps

Table 2.2 MPEG-2 Levels

For standard, broadcast quality coding, the main profile is used in conjunction with the main level. Scalable coding is characterized by the ability construct useful video using only segments of the complete bit stream. The total video signal can be split up into different layers with enhancement layers providing the ability to increase the spatial resolution (spatial scalability) or decrease the quantization distortion of the base layer (SNR scalability).

The sampling matrix refers to the extent of subsampling carried out upon the chrominance components of each image. A 4:2:0 image has both chrominance components subsampled horizontally and vertically by two, a 4:2:2 image has the chrominance components subsampled horizontally by two, and a 4:4:4 has an equal number of chrominance components as luminance components (see figure 2.6) [10]. The 4:2:0 is generally preferably because it is the most compact representation and the amount of image quality loss, compared to the other two formats, is only slightly greater.

2.1.5 Video File Servers

Video file servers will play an important role in the proliferation of digital video, especially MPEG-2 sequences. Although real-time encoding and transmission of visual information is required in certain situations, the bulk of digital video will be found as preprocessed video files in a system of file servers intended for broadcast. Two principal examples are *Video on Demand* (VOD) and *Interactive Television* (ITV). Video on Demand is based upon the strategy of providing customers with a huge selection

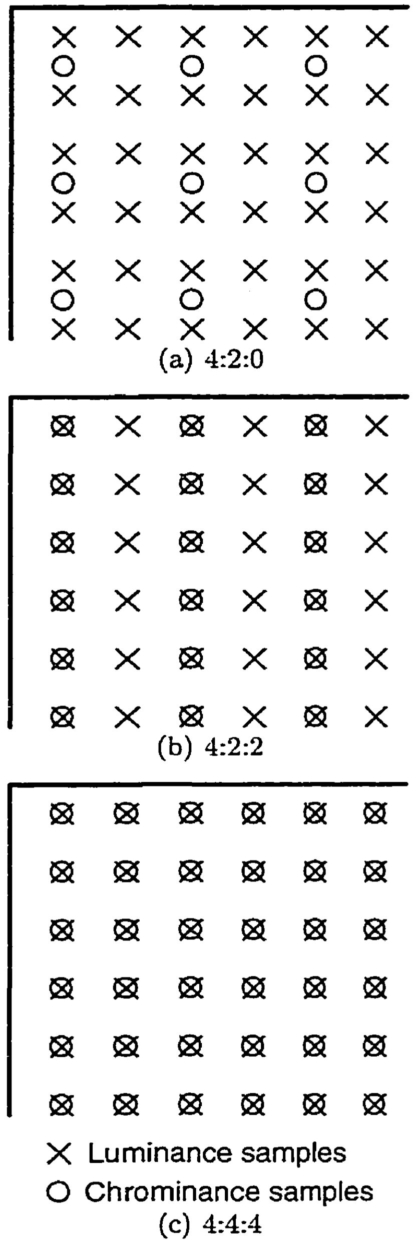


Fig. 2.6 Position of luminance and chrominance components for different sampling matrixes

of movies that may be played within moments of being selected. If the server provides additional "VCR-like" abilities, such as fast forward/reverse and pause, the service becomes known as Interactive Television. Additionally, ITV can include services such as interactive video games. These projects aim to displace video rental outlets with the enticement of greater convenience. Presently, most multimedia applications are found on stand-alone PCs but there is an increasing trend towards using video file servers to store multimedia data. This direction is motivated by reasons of data sharing, security, data integrity, and centralized administration [14].

A single video sequence is usually *striped* (distributed) across several storage nodes in the video file server system. This is done to increase the number of subscribers that may access a particular movie at a given time and also to improve the load balance across the system. Time-division multiplexing is the approach taken when sending a stream to the network. The server's network usage is divided into a number of fixed-sized time slots and each slot is assigned to a different video stream. During a slot, the storage node containing the next sequentially ordered block (a block is unit of striped video data) for that stream will begin transmission across the network. The file server cycles through the other slots and then returns to the current slot a fixed amount of time later (T seconds). This scheme is really intended for transmission of constant-bit-rate streams. More applicable to variable-bit-rate video is the continuous media server developed by Neufeld, Makaroff, and Hutchison [15]. In their version of a video file server, the number of fixed-sized blocks that may be read for each slot is variable. For variable-bit-rate video, this means that during each slot, a sufficient number of blocks to represent T seconds of displayed video data must be buffered and transmitted before the server returns to the current slot, otherwise real-time video may not be attainable at the client.

2.2 Constant and Variable-Bit-Rate Encoding

Traditionally, information has been sent through networks in fixed bandwidth channels. To fully utilize the channel, a source must try to produce a constant stream of bits with a transmission rate equal to the size of the channel. Newer digital networks, such as ATM, provide the flexibility to handle traffic sources which vary their bandwidth requirements over time. Thus, source coders now have the option to transmit their data without expending the effort to confine the bit rate to a single value. Both constant-bit-rate encoding and variable-bit-rate encoding have their merits and disadvantages as the remainder of this section explains.

2.2.1 Constant-Bit-Rate Encoding

As a result of the compression procedure, an MPEG coder produces a variable-bit-rate data stream. However, for many situations, the bit stream will be carried by a fixed-bandwidth channel. This is because transmission over fixed-bandwidth channels is a very well understood and studied problem, particularly in light of the maturity of telephony technology. To enable an MPEG coder to be able to produce a constant-bit-rate (CBR) output, a buffer with a constant service rate must be inserted between the coder and the channel (see figure 2.7). A feedback loop is connected from the

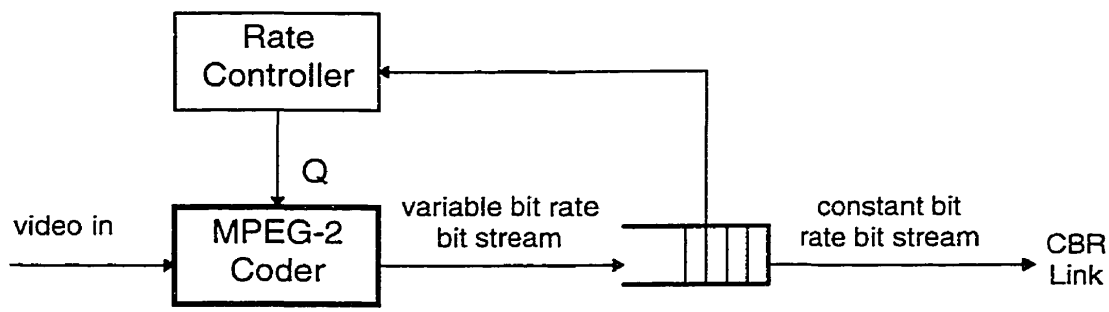


Fig. 2.7 Constant-bit-rate encoder

buffer to the quantizer control of the coder through a *rate controller*. The rate controller's function is to translate the buffer occupancy information into an encoding quantization scaling factor, Q, that will prevent the buffer from overflowing or underflowing. As previously mentioned on page 11, Q is a variable whose value controls the coarseness with which the DCT coefficients are quantized; a higher value results in more coarsely quantized coefficients and a reduction in the number of coding bits, but at the expense of image quality. This modulation of Q over time aids the buffer in outputting a constant bit stream by ensuring that the buffer always has bits waiting to be serviced but never enough to exceed its capacity. Specifically, if the buffer occupancy exceeds a certain threshold, then the rate controller will increase Q to reduce the bit rate. Likewise, if a lower threshold is met, then Q will be decreased. The exact method of the buffer rate control is typically heuristically derived [6]. In MPEG-2, the quantization factor can not only be influenced on a frame-by-frame basis, but also at the *macroblock* level (a macroblock is simply a 2×2 grouping of luminance blocks along with the corresponding chrominance blocks) [10].

MPEG-2 explicitly defines the maximum buffer size that is to be used in the encoder and decoder. Although a larger buffer would be better able to smooth out the bit stream, enhance network statistical multiplexing, and aid in the de-jittering of the received images [16], it introduces a significant amount of end-to-end delay. This will be in addition to the frame reordering delay necessary because of the difference between the order in which frames are encoded and displayed. In the case of real-time video decoding, the constraints on the network delay must be strict in order to prevent information from arriving too late to be useful.

The quality of the decoded images is a direct consequence to the number of bits invested in the encoding process. Besides the adjustment of the quantization scaling parameter, another method to influence the number of coding bits is to change N and M, the group of picture parameters (see page 13). In practice, the quantization adjustment method is preferred because of the ease in adjusting Q and because the changes in the bit rate are effective immediately. To indicate the level of influence that the quantization scaling factor has upon the number of encoding bits, figure 2.8 presents the relationship between the number of bits required and Q for the first frame (intra coded) taken from the "calendar" sequence.

2.2.2 Variable-Bit-Rate Encoding

Variable-bit-rate (VBR) transmission, on the other hand, possess a couple of key advantages over its constant-bit-rate counterpart, namely, the ability to produce constant-quality video sequences, and the facilitation of statistical multiplexing in networks [17].

Perhaps the greater of the two advantages is the ability to maintain a consistent image quality over all of the encoded frames (see figure 2.9). Without the need to accommodate a rate buffer, the quantization level can be kept constant and allow each frame to produce as many bits as needed to maintain the current image quality.

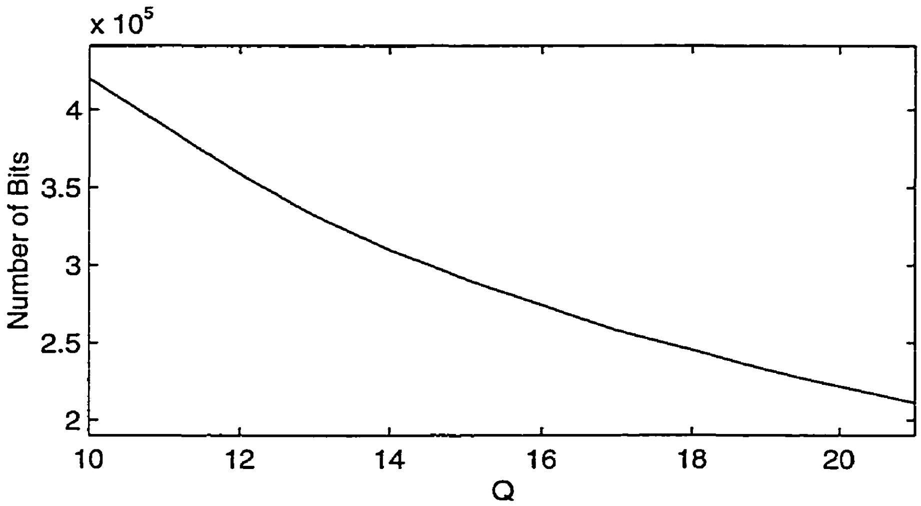


Fig. 2.8 Relationship between the number of coding bits and Q

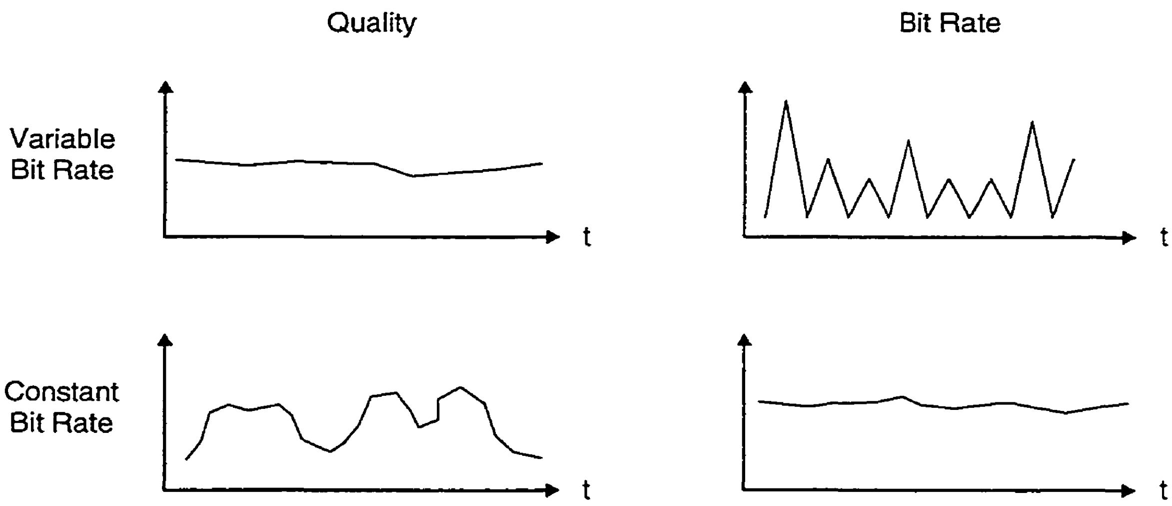


Fig. 2.9 Comparison of CBR and VBR image quality and bit rate characteristics

Typically, the image quality is evaluated by computing the *peak signal to noise ratio* (PSNR) of the luminance and chrominance components of each decoded frame. The PSNR is defined as

$$PSNR = 10 \log \left(\frac{255^2}{MeanSquaredError} \right) dB \tag{2.1}$$

Intuitively, a constant-bit-rate coder can be made to operate in a variable-bit-rate fashion by removing the rate buffer and simply disconnecting the feedback loop in figure 2.7. For this reason, this type of coder is typically referred to as an *open-loop coder* figure 2.10). The absence of the buffer alleviates the problem of introducing

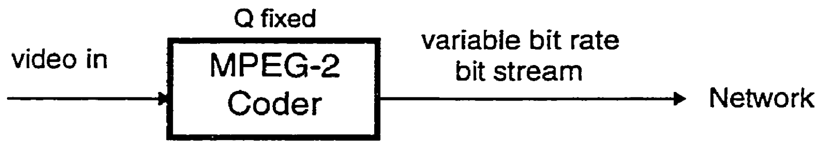


Fig. 2.10 Variable-bit-rate (open-loop) encoder

additional time delays. Furthermore, without the need for a rate controller and rate-control algorithms, the coder design is simplified and, in general, operates faster than a CBR implementation. VBR MPEG-2 bit streams, when decoded, produce videos that are less objectionable to view because of a greater consistency in picture quality. This does not imply that the decoded video will necessarily have a superior average peak signal to noise ratio compared to a CBR encoded video sequence. Explicitly stated, the central goal of variable-bit-rate video coding is to have a constant (or more realistically, consistent) image quality rather than to maximize the instantaneous quality within the boundaries of the resource constraints [18].

I, P, and B frames

Certain considerations exist when attempting to impose a uniform image quality over the entire set of frames in a video sequence. While the encoded information for an I frame is comprised of the actual spatial image information, the information encoded for the interpolated frames consists mainly of the motion information and error signals. As a result, maintaining a constant image quality over all frames (in terms of PSNR) requires that proportional constants be applied to the quantizer when encoding P and B frames to lower their PSNR. In addition, B frames are coded with an even lower PSNR than P frames because they are not used for prediction purposes [11]. If the quantization scaling factors used for intra and predicted frames are Q_I and Q_P respectively, then the proportional constant for Q_P is some number, K_P, such that $K_P = Q_P/Q_I$ and, for B frames, $K_B = Q_B/Q_I$ with $K_B > K_P > 1$. The values for K_P and K_B may differ slightly for different sequences and must be determined experimentally.

Rate-Constrained VBR Coding

The reliable transmission of data with time-varying bit rates is now very possible because of the introduction of ATM in broadband ISDN networks (section 2.3). Furthermore, these networks have the ability to make more efficient use of the network bandwidth through a method called *statistical multiplexing*. This is a very important advantage from the network point of view; VBR connections allow a greater flexibility because they allow the network to dynamically re-allocate bandwidth to improve the efficiency with which the network resources are used. However, the time-varying property of the bit-rate characteristic of open-loop encoders pose a major difficulty to networks. Unlike the case for CBR traffic, VBR traffic cannot be assigned a fixed bandwidth without either wasting an enormous amount of network capacity (bandwidth over-allocation) or losing some information during high bit rate periods (bandwidth under-allocation). Instead, rather than allocating a fixed-bandwidth channel, the network allows a set of bit-rate descriptors such as the peak bit rate and the average bit rate. The conformance to these descriptors ensures that the variable-bit-rate traffic will arrive at the destination under a set of quality guarantees provided by the network. To conform to the network's demands, a different class of VBR coders, called *rate-constrained* or *traffic-shaping* encoders, are used (figure 2.11). These encoders generate variable-bit-rate data, although the variability of the bit rate is subject to a number of constraints. Most traffic-shaping encoders use a policing function, similar to ones employed by networks, to monitor their output bit rates and make changes when necessary to maintain conformance with the network traffic descriptors. Comparing figure 2.7 and figure 2.11 reveals that if the policing function performs the same task as a buffer and rate controller, then the VBR coder can be made to mimic

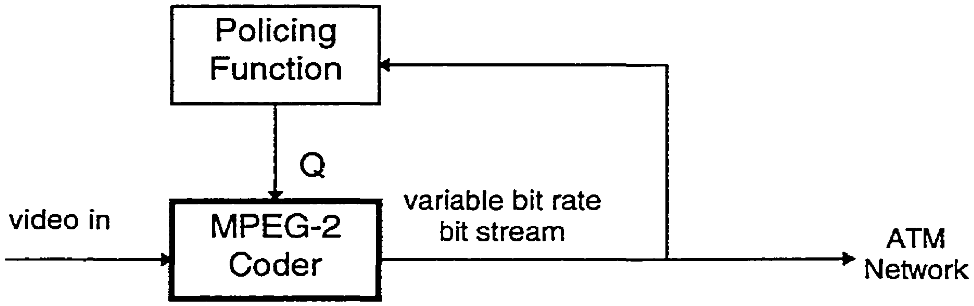


Fig. 2.11 Traffic-shaping variable-bit-rate encoder

a CBR coder. In fact, constant-bit-rate encoding can be thought of as a special mode of operation of a variable-bit-rate coder.

The method of rate control is largely an open problem and is not only limited to bounding the rate envelope. Others have, instead, tried using a stochastic model to describe the time-varying rate. For instance, Heeke [19], using a Markovian model for variable-bit-rate video traffic, describes a rate-control algorithm that is effective provided that certain limitations to the transmission rate are maintained. Unfortunately, most models are difficult to enforce and verify, thereby limiting their use to the network connection phase (call acceptance control) [20, 21].

An elaboration of the topics of ATM networks and network policing will be made in the next section.

2.3 ATM and Packet Video

Digital video is the most demanding application in terms of network transmission bandwidth. The broadband networks that will be capable of carrying a multiple number of digital video sources, such as the *Broadband Integrated Services Digital Network* (B-ISDN) will almost certainly rely on the *ATM* transmission protocol.

ATM, short for *Asynchronous Transfer Mode*, is a network transmission protocol devised to support and integrate a diverse set of applications, regardless of their bit-rate statistical characteristics [22]. In addition, ATM networks very efficiently exploit the available bandwidth by statistically multiplexing the different sources while preserving the integrity of the transmissions by providing quality of service (QoS) guarantees.

ATM is particularly well suited to digital video traffic because of its capability to seamlessly handle both constant and variable-rate sources along with its ability to handle high bit rates. This flexibility stems from the asynchronous transfer of small, fixed-length packets of information. The versatility in being able to handle a variety of incongruent traffic has created the requirement for a "policing" function to monitor the traffic and ensure that each data stream is behaving as expected and not adversely affecting any other transmissions.

The ensuing discussion gives a very general overview of the ATM protocol. Special mention is provided on the network traffic handling and the relationship between ATM and digital video. For a more comprehensive description of ATM and B-ISDN, many texts, such as Onvural [23], Ohta [1], and Bertsekas and Gallager [24], are available.

2.3.1 Overview

The idea of transmitting data in small packets over asynchronous transfer channels originated from experiments conducted in the late sixties [8]. The desire to be able to transmit not only data, but simultaneously, a combination of voice and video (multimedia) has fueled the research that has resulted in the development of B-ISDN and ATM.

The Broadband Integrated Services Digital Network is envisioned as the network of the future. With its enormous bandwidth and flexibility, it will be an all-encompassing network able to simultaneously carry an impressive number of heterogeneous digital transmissions, including full-motion video signals. On the basis of its numerous strengths, ATM has been chosen by many standards organizations (including ANSI and ITU) to be the underlying transport technology within B-ISDN [25]. In the context of B-ISDN, the functions associated with switching and multiplexing are referred to as the *transfer mode.*

The Asynchronous Transfer Mode combines the circuit switched routing of telephony networks with the asynchronous multiplexing of packet switching. With respect to the Open Systems Interconnection (OSI) model, the ATM layer replaces the bottom two layers (the physical and data link layers) and part of the third layer (the network layer). ATM can essentially be thought of as a scheme for packaging information into a unit called a *cell.* Each cell is fifty-three bytes long, including a five byte header which contains the routing instructions among other information. The difference between ATM and traditional packet switching technology is that ATM assumes high-speed, low-error circuits (i.e., fiber optics). Thus, ATM is relieved of the burden of carrying a complex retransmission control. Furthermore, unlike packet switching, which is essentially a switching technology, ATM handles the switching along with the transmission linkage [1].

A fixed route (virtual circuit) is established before accepting any traffic request (known as a call). Traffic flowing through a link in the network will consist of an interleaved mixture of cells originating from different sources. The network will ensure that all cells belonging to a single source follow identical paths. The significance of this is that it allows the network to offer *quality of service* guarantees, specifically, the cell loss rate, the maximum delay, and the maximum delay variation. This ability to provide QoS guarantees is the major aspect of ATM that differentiates it from other network transfer technologies such as Frame Relay and the Internet Protocol.

The network must be advised about the particular behavior that is to be expected from a source in order to be able to offer delay and loss guarantees. The process by which the network evaluates and determines the source's suitability for admission into the network is called *connection admission control* or simply CAC. The decision to accept or reject a source is based upon the traffic descriptors provided by the source. Figure 2.12 reveals the traffic management process. The enforcement of these traffic descriptors is the responsibility of the network policing function.

2.3.2 Network Policing

There are several reasons why a client may provide inaccurate traffic descriptors. An oversight or an overly conservative estimate of the bit stream's behavior may simply be the case. Another motive may be that the client wishes to reduce the transmission cost and intentionally specifies inaccurate parameters. Still another possibility is the malicious intent to overload the network. In any case, a network-source contract must exist that will impose constraints upon the source for any realistic VBR scenario.

In conventional fixed-rate transmission, only the disclosure of the peak rate was required for network design and service provisioning. However, variable-rate transmission is a much more difficult situation in terms of network management and requires

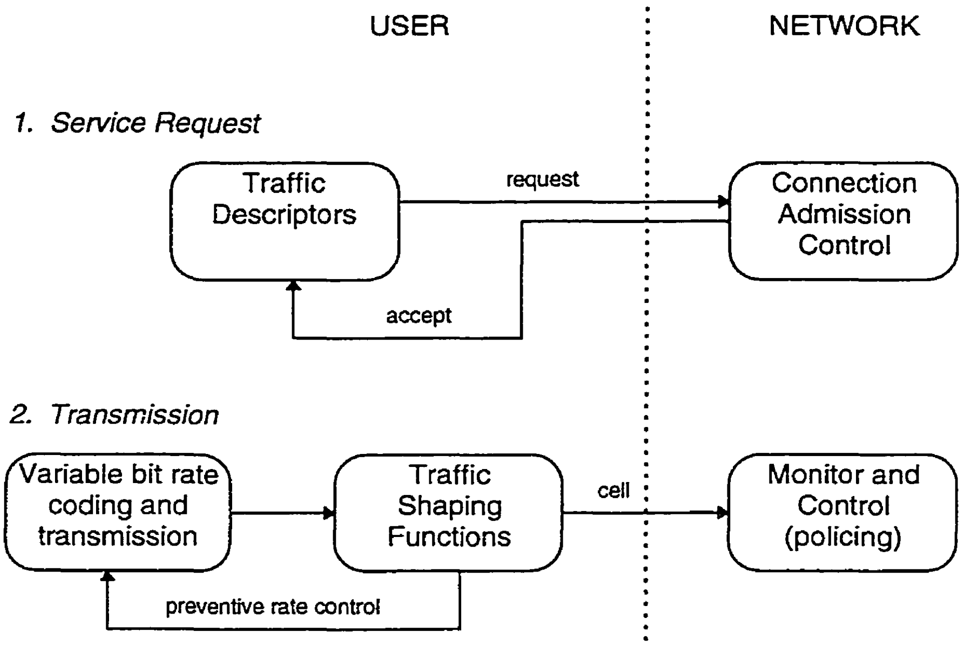


Fig. 2.12 Traffic management process

the declaration of a more complete set of descriptors. The two most logical choices for descriptors are the peak rate and the sustained, or average, rate of the source. However, with only these two parameters, the opportunity exists for the data stream to transmit at the peak rate for a prolonged period of time without violating the negotiated sustained rate. This situation can be dangerous to the network as, for VBR transmissions, if several traffic sources sharing the same link should momentarily elevate to a higher bit rate at the same time, packet loss and/or increased delay could occur. Therefore, a third descriptor, namely, the *maximum burst size* or the *burst length* is required. The burst length specifies the maximum duration that the source may continuously transmit at the peak rate. In the ATM standardization, the rate control function based on these traffic descriptors along with the policing operation is call the *Usage Parameter Control.*

The International Telecommunication Union (ITU) and the ATM Forum (a consortium of leading computer communications manufacturers and end-users) are currently in the process of standardizing the Asynchronous Transfer Mode. In their User Network Interface (UNI) [25], the ATM Forum has declared that the definition of the network traffic conformance to the traffic contract be based on the *Leaky Bucket* policing function. The Leaky Bucket is the most prevalent constraint system used and has the advantage of being very simple [18].

The Leaky Bucket

The Leaky Bucket algorithm may be modeled as a $G/D/1/N$ queue [26]. Heuristically, the scheme can be thought of as a fixed-height "bucket" that is being filled by the incoming packets while continuously leaking at a fixed rate, L (figure 2.13). A counter is used to monitor the level, or occupancy, of the bucket. Each arriving

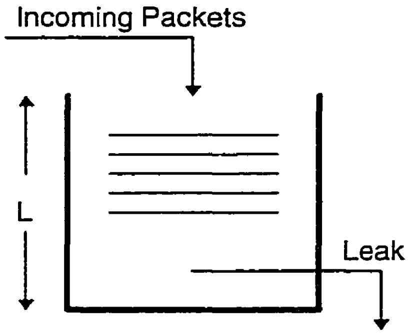
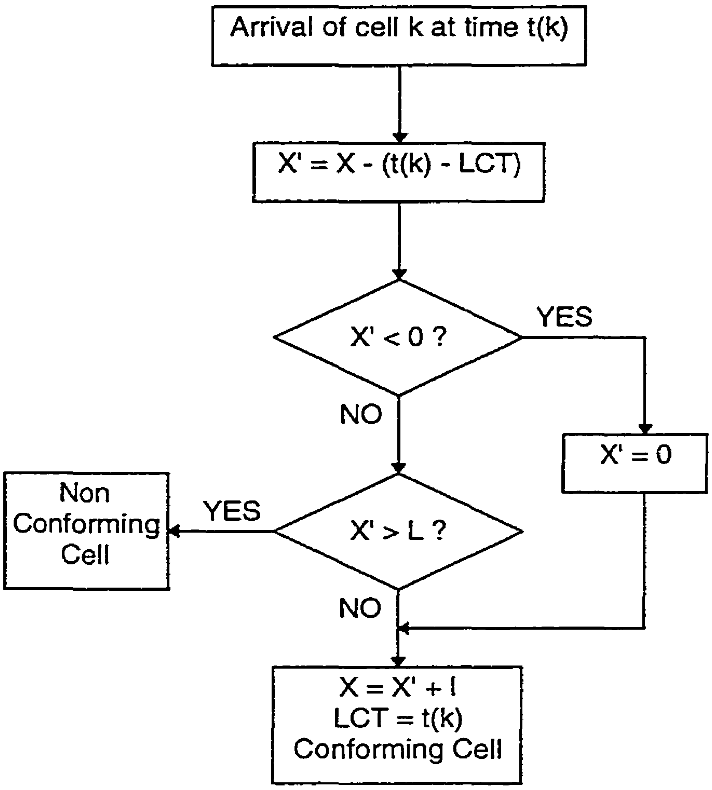


Fig. 2.13 Conceptual Leaky Bucket

packet increments the counter and if the value exceeds the bucket size, then the bucket "overflows." In this situation, the network can either mark the overflowing cells as low priority, or more drastically, discard them [27]. In actuality, no cells are channeled to a bucket, rather, the bucket is an imaginary buffer used to count a stream of cells as they pass through the link.

The Leaky Bucket algorithm, also known as the *Generic Cell Rate Control Algorithm* (GCRA), is formally defined by the flow chart shown in figure 2.14 [25]. The bucket level increases by I for each conforming cell and decreases at a continuous rate of one unit per time-unit. The GCRA is dependent upon two parameters only: the increment, I, and the limit, L and this is reflected in the notation GCRA(I, L). The capacity of the bucket (the upper bound of the counter) is $L + I$.

In practice, two Leaky Buckets are necessary to enforce the traffic descriptors; one for the peak rate and another for the sustained rate and the burst length. Supposing that the peak cell rate (PCR), the sustained cell rate (SCR), and the maximum burst size (MBS) are specified as R_P, R_S and mbs respectively, then the Leaky Bucket



X	Value of Leaky bucket counter
X'	Auxiliary variable
LCT	Last Compliance Time
I	Increment
L	Limit

(X = 0 and LCT = t(k) for the first cell.)

Fig. 2.14 Generic Cell Rate Control Algorithm needed to monitor the peak transmission rate is GCRA(T_P, 0) and the one for the sustained transmission rate and burst size is GCRA(T_S, τ_S) [25] where

$$\begin{aligned} T_P &= 1/R_P & (2.2) \\ T_S &= 1/R_S & (2.3) \\ \tau_S &= (mbs-1)(T_S - T_P) & (2.4) \end{aligned}$$

The exact meanings of the sustained cell rate and the maximum burst size in the context of the GCRA deserve some clarification. Similar to the peak cell rate, the SCR is only an upper bound on the possible average rate of the ATM connection, and is not a target rate. This has an implication to traffic shaping coders that use the Leaky Bucket (clarified in Chapter 4). Next, the defined MBS does not imply that bursts of this length can occur any arbitrary time later; the algorithm of the leaky bucket dictates that the traffic must maintain a lower cell rate long enough for the SCR bucket to empty. Also a consideration is that a larger specified burst size has the ramification that, by increasing the size of the bucket, the length of time over which the average rate is enforced is longer which allows for more variability in the cell rate.

Despite its advantages, the Leaky Bucket algorithm has come under some criticism because it is unable to recognize and monitor any other traffic characteristics. In addition, Butto et al. [26] have questioned the ability of the Leaky Bucket to control mean rates and burst lengths in certain situations. Perhaps it is for this reason that the ATM UNI makes provisions for allowing the use of other traffic monitoring algorithms. Although the traffic conformance is defined in terms of the GCRA, the network provider may use any other Usage Parameter Control as long as the QoS guarantees are maintained [25].

2.3.3 VBR Traffic and Statistical Multiplexing

As mentioned in the previous section, variable-bit-rate traffic enables the network to dynamically redistribute network resources and thereby increase the network efficiency. More precisely, the bandwidth allocated to each VBR source will be less than its peak, but necessarily greater than its average bit rate. Statistical multiplexing is achieved when the sum of the statistical bandwidths of the multiplexed connections on a link is less than the physical bandwidth even though the sum of their peak rates exceeds the link bandwidth. Statistical multiplexing is most effective when the required bandwidth for each individual source is much smaller than the total link capacity which is shared by a number of different sources [18].

Another benefit of VBR coding in an ATM environment is the reduction of end-to-end delays [22]. Without the substantial delays introduced by the large end buffers of CBR sources, information is, more or less, transmitted to the network as it becomes available. While this property is an important one to all types of digital video transmissions, it becomes a crucial feature for interactive connections, such as video conferencing.

2.3.4 Issues with Digital Video

The asynchronous transfer of video is often referred to as *packet video* and is defined as the transfer of video signals over asynchronously time division multiplexed networks such as ATM and IP [20, 21]. Packet video can provide a higher quality of service in terms of flexibility, rate variability, and constant quality through variable-rate transmission [1]. However, there are a couple of notable difficulties to contend with, especially within the context of MPEG.

The joint problem of traffic characterization and rate control becomes a key issue in packet video. Despite numerous attempts to model the source video characteristics [19, 28], none have been successful enough to universally apply to all video sequences. Consequently, the most often used method to determine the traffic descriptors of a sequence is to simply encode it using an open-loop encoder and then to analyze its bit rate profile [18].

Network errors can pose a problem for MPEG data streams. The cell loss rate and the delay jitter have been identified as the major problems for transmissions of VBR video traffic over ATM. Of the two, the cell loss seems to be the bigger hindrance because of the high data rate nature of video signals. In the case of MPEG, a particular problem arises because of the I, P, and B frame coding structure. As explained previously, the I frames tend to achieve the least amount of compression and therefore require the greatest number of bits to encode. This results in abrupt increases in the bit rate every time an I frame is transmitted. During such high transmission rate periods, the bit stream is more susceptible to network errors because if other bit streams concurrently experience elevated bit rates, there will be a large possibility that the aggregate instantaneous bit rate on the link will exceed the link's actual bandwidth resulting in buffer overflows at the nodes, or else, additional transmission delays will occur. Errors in I frames are particularly harmful because I frames serve as anchors for an entire group of pictures and thus, any information loss can adversely affect all subsequent pictures until the next I frame.

Chapter 3

Variable Bit Rate Video Encoding

The initial focus of this thesis research revolves around variable-bit-rate video encoding. VBR source coding affords key advantages to both the user and the network. As explained in the preceding chapter, permitting the number of encoding bits to vary between frames results in a more consistent picture quality in the decoded sequence. Moreover, when several of these bit streams are concurrently transmitted, the network has the potential to statistically multiplex them together to more effectively utilize the available bandwidth.

In the ensuing section, an outline of the steps taken to transform a constant-bit-rate MPEG-2 coder into a variable-bit-rate version is given. The remainder of the chapter is devoted to presenting and examining examples of variable-bit-rate encoded bit streams and comparisons are made to constant-bit-rate equivalents.

3.1 The INRS MPEG-2 Encoder

The MPEG-2 coder considered in this study is a software implementation developed by Nadia Baaziz, a researcher at the Institut national de la recherche scientifique (INRS) Télécommunications. While it does not possess all the functionality specified in the international standard, it is capable of coding sequences at the main level and main profile. The INRS coder operates at constant bit rate and follows the rate-control algorithm outlined in the MPEG-2 Test Model 5 [29].

The Test Model 5 (TM5) is the best known, published coding model of the MPEG-

2 algorithm and was used during the evaluation and verification stages of the standard. An encoder based on this model will produce bit streams that possess a constant bit rate. More precisely, the encoder maintains a fixed bit rate between groups of pictures (GOPs) by using a bit allocation scheme to appropriately vary the allowance of bits between frames and even within each frame. This rate control mechanism accomplishes this by using feedback information provided by the rate buffer to judiciously modify the quantization step level (Q) of the coder. The structure of a TM5 encoder is essentially that of figure 2.7.

Naturally, with a software realization, the goal is to produce a data file for storage onto a file server system to be transmitted at a later time rather than for immediate transmission. The obvious advantages of this are that equipment capable of real-time MPEG-2 coding is unnecessary (i.e., hardware coders) and that the MPEG sequence can be re-transmitted multiple times without the need to recode the video at every instance.

3.1.1 Modifications to the Coder

The conversion of the INRS coder to produce a variable-bit-rate output is conceptually as simple as disconnecting the feedback loop from the buffer to the coefficient quantizer (figure 2.10). This is, in fact, exactly what is done to the actual coder. By bypassing the routine calls to the rate controller after processing each macroblock, the quantization step level is forced to remain fixed for the duration of the sequence and the resulting MPEG-2 bit stream will have a variable rate.

Besides circumventing all function calls to the rate controller, other minor modifications to the program code are necessary to achieve VBR operation. The MPEG-2 international standard, being a broad standard, makes concessions for an MPEG-2 coder to operate in VBR mode. One specification is that the Video Buffering Verifier (VBV) delay be set to the hexadecimal value *FFFF*. The VBV is a hypothetical decoder with an input buffer called the VBV buffer, and their joint purpose is to generate buffer occupancy information that will assist the decoder in decoding the bit stream [10].

For an open-loop VBR MPEG-2 coder, the end buffer would appear to be made an extraneous accessory by the absence of a rate controller. Although the removal of the buffer would free up a potentially large amount of computer memory, it remains an important part of the modified INRS coder because it speeds up the transfer of data to the disk.

3.1.2 Fixed Quantization Scale Factor

Unlike the method of the Test Model 5, the quantization scale factor for the modified MPEG-2 coder is held constant over an entire frame. The motivation behind this is to maintain a uniform image quality level over at least a complete frame. In this case, any change in picture quality will be observed between frames. In conjunction with a fixed value for Q, suitable values of K_P and K_B are needed to maintain a steady PSNR level across the different MPEG frames. Through experiments using the "bus" sequence, the values of $K_P = 1.45$ and $K_B = 1.82$ were found to provide very good results for the VBR case. Furthermore, these values proved to be suitable for use with the other sequences.

3.2 Experiments and Results

The sequences used in the experiments are titled "bus," "flower garden," and "calendar." These video clips were chosen because they all posses a varying amount of motion and a significant amount of image detail which results in bit rates that span a very wide range. The first frame of each sequence is displayed in figures 3.1 to 3.3. All the sequences used by the coder are 4:2:0 image sequences with picture dimensions of 704 by 480 pixels. (Recall that a 4:2:0 image is one that has both its chrominance components vertically and horizontally subsampled by two.) Unlike a completely functional MPEG-2 coder (as specified in the standard), the INRS coder is only capable of manipulating input sequences that meet these specifications.

After the three sequences have been encoded and decoded by the VBR MPEG-2 codec, the profiles of their bit rates and time-varying image qualities are analyzed. To be able to reveal the merits of variable-rate encoding, constant-bit-rate encoded equivalents are needed for comparison. However, obtaining suitable examples proved to be a rather difficult undertaking.

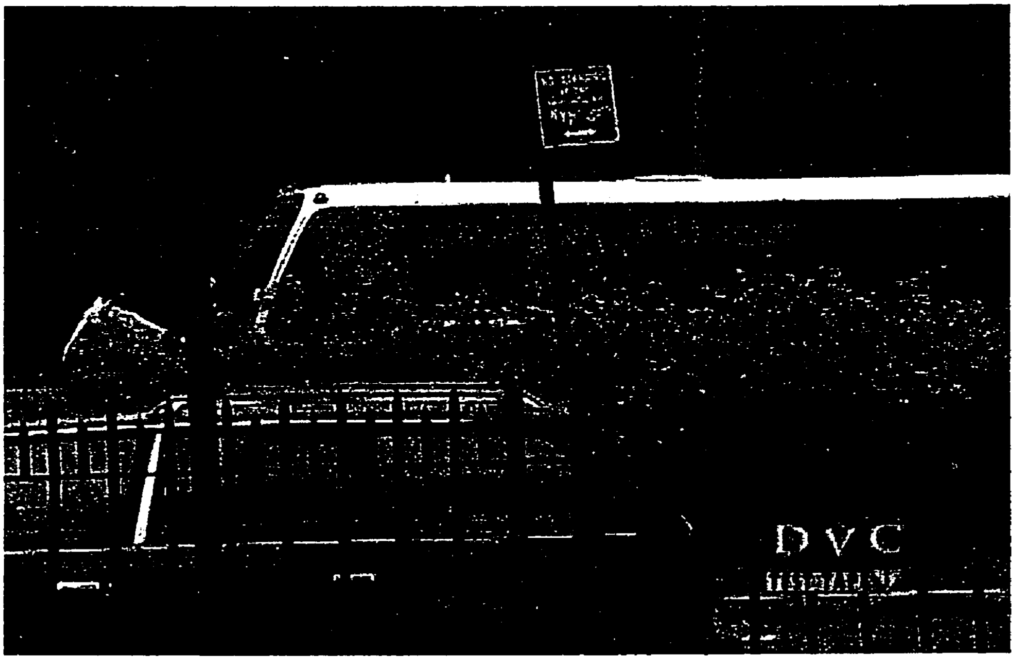


Fig. 3.1 Frame 0 for sequence "bus"

Fig. 3.2 Frame 0 for sequence "flower garden"

Fig. 3.3 Frame 0 for sequence "calendar"

3.2.1 Achieving Constant Bit Rate

Although the TM5 encoder is able to output a CBR bit stream, the uniformity of the bit stream is, unfortunately, only realized if the bit rate is observed between entire groups of pictures. For the typical values for the frame rate of thirty frames per second (30fps) along with a GOP length of fifteen frames, a constant bit rate is perceived only when observations are made in half second intervals. This time length may, in some cases, be deemed excessive for practicality, and for the purposes of this study, a more refined constant bit rate coder is desired. By realizing that CBR operation is a specialized case of variable-bit-rate coding, the rate-constrained coder (discussed in detail in Chapter 4) can be used to produce a constant-bit-rate output. This is basically accomplished by equating the target values of the peak bit rate and sustained bit rate.

An important consideration in maintaining a uniform bit rate is the period over which the uniformity is evaluated. In other words, what will be the time interval over which the average bit rate will be calculated? For the off-line software coder, the only reference of time available is the frame rate, f (i.e., a period equal to $1/f$

will have elapsed during the transmission of the complete set of bits representing a single frame). As a result, one frame period will serve as the basis for the interval of observation. The interval of observation of n frame periods is another degree of freedom available for constant bit rate compression. A value of n greater than one offers the advantage of allowing the bit rates for the individual frames within the group to naturally vary as long as the overall average remains consistent. For MPEG-2, this is especially advantageous because of the I, P, and B frame system. Since I and P frames are almost always followed by a B frame (see figure 2.4), and B frames generate significantly fewer bits, a greater value of n will allow the desired average bit rate to be achieved without drastically reducing the bit rate of an I or P frame. This implies that no one frame will be required to sacrifice a significant number of bits and suffer a noticeable quality degradation. Instead, the impact of the bit-rate confinement is absorbed by all n frames in the group.

The parameter n has a practical importance when considering a video-file-server system. A file server typically places into its buffer video data representative of a predetermined amount of time. If this data encompasses several frames, their individual bit rates will be averaged by the constant service time of the file server's buffer. To take advantage of this, the encoder can specify n to match the value used by the server's buffer to generate the best quality possible for a particular bit rate.

3.2.2 Practical Considerations

Certain concessions exist for operating the rate-constrained VBR coder to produce constant-rate traffic. Since the central motivation for constant-bit-rate coding is to produce a bit stream where the number of bits per transmission interval is as constant as possible, the quantization scale factor is allowed to change dramatically from one series of frames to the next to attain this goal. This implies that there is no graceful image quality change between consecutive frames. A graceful quality change would be visually less objectionable, but the bit stream would deviate further from constant-bit-rate operation. When the traffic shaping encoder is used to produce a constant-bit-rate output, the maximum burst size, mbs, becomes an irrelevant parameter. As seen from equation 2.4 on page 29, the mbs parameter influences the size of the sustained bit rate (SBR) Leaky Bucket. However, since the leak rates, T and T_S, will be equivalent,

τ_S will be zero, regardless of the value of *mbs*.

Contrary to what many authors advise, a constant bit rate cannot be achieved by simply equating the peak and sustained bit rates to the desired bit rate in the rate-constrained coder. In defining the sustained bit rate, the Leaky Bucket algorithm only attempts to prevent this value from being exceeded. In practice, this upper limit is never achieved. The explanation for this is that for the bit rate to average to a certain value, it must be allowed to fluctuate above and below that value. However, with the peak-rate Leaky Bucket imposing a restriction upon the maximum possible bit rate that is exactly identical to the sustained rate, the average bit rate will necessarily be below the desired rate. Because of this limitation, the specified sustained and peak bit rates have to be assigned a value greater than the desired output average bit rate. The exact amount of increase is not exactly predictable but usually lies within the range of 200 to 500 kbits/s for the bit rates considered in this study. In addition, to enable the frames, or group of frames consisting mainly of B frames to be able to attain the target bit rate, the initial value of Q must be lowered dramatically. The degree with which Q is lowered is dependent upon the value of n used, with higher values of n requiring a larger decrease in Q from the value used in the pure VBR encoding process. A special situation exist when the bit rate average interval, n, equals one frame. In this case, no chance exists for smoothing the bit rate between consecutive frames and each individual frame must abide by the strict limitations enforced by the rate controller. This means that I and P frames will have their bit counts noticeably cut while the B frames will have to have their bit rates raised abnormally high. To facilitate this end and speed up the encoding process, the initial value of Q can be increased by around five more than the open-loop value and the values of K_P and K_B can be adjusted to lower values such as 1.00. This effectively encodes each of the P and B frames with abnormally high initial bit counts (since the lower values of K_P and K_B translates to a lower values of Q) so that the coder does not have to perform too many time-consuming adjustments to equalize the frames sizes.

Because the adjustments to the quantization scaling factor are made on a frame by frame basis, it is extremely difficult to maintain a perfectly constant bit rate over the entire range of the frames. In fact, a single-step increment or decrement can cause the number of encoding bits to change by thirty-thousand bits or more. As a result, the bit rate from one series of frames to the next may oscillate around the overall average. However, the variations will be minor compared to the open-loop case and it is assumed that the file server's buffer will provide additional smoothing to the bit stream.

3.2.3 Software and Hardware

To create the bit streams necessary for this study, two versions of the modified INRS MPEG-2 coder are used. The first one is the one described in this chapter and produces the unconstrained VBR compressed video signal; the second is the further modified coder possessing traffic-shaping functions and is introduced in the next chapter. By trying various traffic-shaping parameter configurations, the coder can produce constant-bit-rate MPEG-2 files for comparison to the VBR examples. In every instance of encoding, the values $M = 15$, $N = 3$, and $frate = 30$fps are used. The decoding of the compressed files is accomplished using Eckart and Fogg's software MPEG-2 decoder [11]. This decoder is used because of its speed advantage over the INRS developed decoder. For viewing the decoded video sequences, a Viewgraphics Viewstore 6000 image media manager is used to store the sequences and send the images to a Sony broadcast interlaced monitor. With the Viewstore 6000, several video sequences can be viewed concurrently and the individual frames or fields may be examined.

3.2.4 Comparison of Bit Rates

The goal of the experiments performed in this section is to verify that the INRS MPEG-2 coder has been successfully modified to perform variable-bit-rate operations. The first sixty-one frames of the three aforementioned sequences are first coded by the new VBR MPEG-2 software coder and then by the traffic-shaping coder under suitable parameters to coerce it to operate under constant bit rate. Afterwards, the plots of the resultant time-varying bit rates are shown followed by the corresponding PSNR levels for the decoded sequences. For the unconstrained case, the value of the quantization scaling factor is varied to emphasize the relationship of Q with the bit rate and the image PSNR. These results are compared side by side with appropriate constant-bit-rate results. Two methods of producing CBR bit streams are explored.

First, by using the traffic-shaping MPEG-2 coder, results for different values of n are generated to illustrate the effects of varying the averaging interval upon the resulting bit stream. The other method uses the original INRS coder which follows the example of the MPEG-2 Test Model 5. In the ensuing discussion, "CBR" will refer to the constant-bit-rate results or operation of the rate constrained coder; any reference to the Test Model 5 (TM5) will be stated explicitly. For continuity of text, all the plots and their corresponding tables are found at the end of the chapter.

The bit-rate profiles for VBR encoded video are provided in figure 3.4 (page 46) with their statistics tabulated in table 3.1 (page 50). Their CBR counterparts are presented immediately afterwards in figure 3.5 and table 3.2 for the traffic-shaped case and in figure 3.6 and table 3.3 for the TM5 results. In the figures, the frame number on the horizontal axes correspond to the order in which they were encoded rather than to the indices of the temporal position they occupy. In terms of temporal position, the frame axes should be labeled 0, 3, 1, 2, 6, 4, 5, etc.. The bit rates are calculated by multiplying the number of encoding bits per frame by the frame rate.

As expected for the VBR encoded sequences, the smaller values of Q are reflected by the upwards shifting of the plots. The level of variability in the bit rate is often evaluated using a metric called the *burstiness* which is simply the peak bit rate divided by the average bit rate. The values for the burstiness as well as the standard deviations of the bit rates are included in the tables to provide an indication of the level of variability in the required bandwidth. A burstiness close to unity is characteristic of constant-bit-rate traffic while a value closer to two is suggestive of variable-bit-rate traffic.

The efforts of the rate controller are immediately evident by comparing the plots of figure 3.5 with the pure VBR plots, especially when considering that the vertical scale of the graphs in figure 3.5 is roughly two to three times smaller; had the axis scales been similar, the CBR plots would appear almost flat. The case of $Q = 8$ is chosen as the basis for comparison when studying constant-bit-rate coding because an average bit rate close to 4Mbps is a typical operating point of an MPEG-2 encoder. The CBR examples that are used in the comparisons are the ones that are found to provide the closest approximation to the average bit rates of the VBR sequences. In a group of n frames, a change in Q will have an effect on all the frames. Therefore, an increment or decrement of just one level can cause a significant shift in the average bit rate for the n pictures. This is the reason that the bit rate profiles in fig 3.5 exhibit a step-like characteristic. As such, it is improbable that bit rate for the n frames can be molded exactly to a pre-specified rate. Consequently, $n = 1$ provides the most uniform bit rate, as reflected by the lower burstiness and standard deviation, because it allows the coder to adjust the bit rate frame by frame. In the situation where n is greater than one, the bit rates are averaged over each group of n consecutive frames before being plotted to reflect the bit rates that will be observed by the network. Prior to this averaging operation, the bit-rate profiles would assume the forms indicated in the plots of figure 3.7. Clearly, the number of bits used to encode each individual frame is given more freedom than for the $n = 1$ case. Comparing this last figure with the plots for the unconstrained bit rates, it becomes apparent that as n increases the shape of the bit rate curves approach that of the VBR curves.

On the basis of its longer averaging interval and its ability to adjust the bit allocation at the macroblock level, the TM5 is able to produce very steady bit rate (figure 3.6). Because the sequences considered are sixty-one frames long and a GOP of fifteen frames is used, the last frame will be considered alone without any other frames with which to average its bit rate. The last frame that is encoded in each sequence (temporally, frame 59) is a B frame which explains the sharp dip in the bit rate at the end of each plot. Another peculiarity of the TM5 sequences is that the first group of pictures exhibit a bit rate approximately 500Mbps greater than the target bit rate. This can be attributed to the first GOP containing one more I frame than the other GOPs. The TM5 rate controller appears to consider the first M frames as they are entered into the encoder rather than the order in which they are processed. This elevated initial bit rate may cause a problem when the bit stream is transmitted through a network that will not tolerate an increased bandwidth consumption for a period as long as the GOP. However, after the initial GOP, the bit rate seems to settle down and remains very stable around the desired rate. As a result of these two peculiarities, the bit-rate statistics indicate that the Test Model 5 results are burstier than the bit streams obtained by using the traffic-shaping coder. Had the length of the sequence been considerably longer, the values for the burstiness and bit rate standard deviation of the TM5 data would have been lower. Once again, the provided data rate curves have their individual frame bit rates averaged over an interval (fifteen frames for the results discussed here) to indicate the actual transmission rates that will be perceived by the network. Included in each graph of figure 3.6 are plots indicating the variations of the frame bit rates within each GOP. These secondary curves bear a strong resemblance to the VBR curves of figure 3.4. This similarity foreshadows the comparable image quality performance of the two sets of video sequences.

3.2.5 Comparison of PSNR

The problem with the CBR operation of the rate constrained coder is that during normally high bit-rate periods (I and P frames), the number of bits will be cropped, thus decreasing the picture quality whereas during low bit-rate periods (B frames), the bit rate may be exorbitantly inflated causing the picture quality to increase beyond the target level.

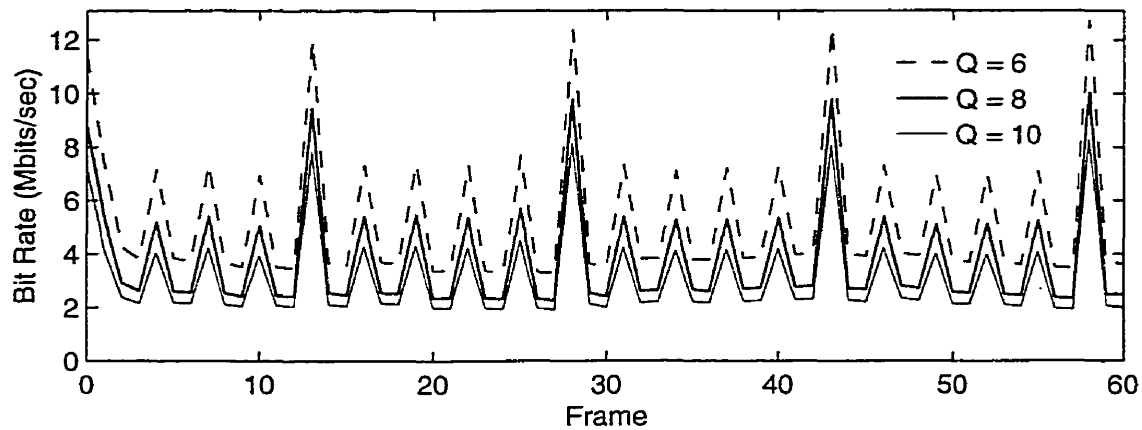
Before the decoded sequences have their image qualities evaluated, they are first converted into ITU-R 601 format which is an interlaced format using 4:2:2 image sub-sampling [2]. This practice is common in evaluating video signals because it facilitates the comparison of the individual fields. Also common is the use of the *peak signal to noise ratio* (PSNR), defined by equation 2.1, as the quality criterion. The PSNR profiles for the unconstrained VBR encoded video sequences are plotted in figure 3.8 with the accompanying table (table 3.4) found on page 54. The PSNR results for the all three video components (luminance, chrominance-1, and chrominance-2) are calculated. However, the luminance is the most important component as it is the chief determiner of the perceived image quality. Therefore, in the ensuing discussion, the luminance will be the focus of examination. The chrominance data is placed in appendix B (graphs) and appendix C (tables).

The first observation that can be made about the PSNR in the different coding situations is that in the variable-bit-rate case, the PSNR is considerably more stable, especially when comparisons are made to the CBR plots for $n = 1$. A steady PSNR is the defining characteristic of variable-bit-rate video coding. Also, just as for the bit rate, a decrease in Q has the effect of shifting the PSNR curves upwards. Upon viewing all three VBR video sequences, very minute or no fluctuations in the time-varying image quality are perceived. In addition, the observed average image quality seems to increase subtly as Q decreases. A value of $Q = 6$ provides a very clear picture while $Q = 10$ presents images with a noticeably grainier appearance and even some blocking effects are sometimes visible. With $Q = 8$ the video images appear to have only slightly more picture noise than for $Q = 6$. A second observation regarding the PSNR is that VBR coded sequences exceed their CBR equivalents in every facet of PSNR statistics (peak, average, minimum, variance). In particular, the average luminance PSNR for the CBR sequences are about one to two decibels lower than the VBR references. This gap seems to decrease slightly as the averaging interval, n increases. When compared to the TM5 results, the VBR sequences still prove superior in terms of PSNR stability but the average PSNRs are very similar. However, it must be observed that the TM5 coder requires a greater bit rate compared to the VBR coder to compress the video sequences. For the "calendar" sequence, the TM5 overshoots its target bit rate by 150kbps which might explain why the average PSNR value is greater than the open-loop version. Upon visual inspection of the Test Model 5 decoded sequences, no differences are perceivable between these and the open- loop VBR sequences.

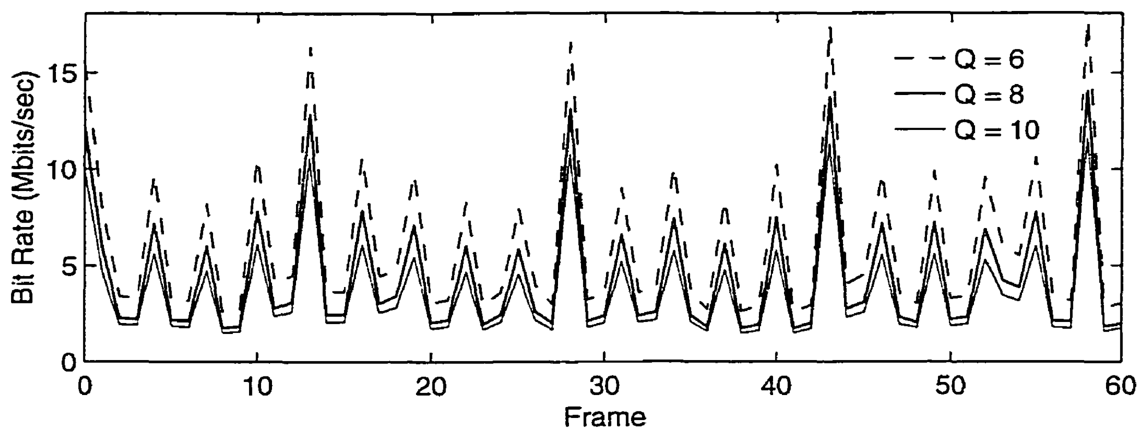
In CBR coding, a larger value of n usually translates to an improved overall picture quality as the degradation is shared among more frames. The plots for $n = 1$ in figure 3.9 illustrate this point. When $n = 1$, each individual frame must be traffic shaped to meet the target bit rate. Intra frames, which rely on a large number of bits to adequately encode their spatial information, must sacrifice their quality considerably to conform; they are indicated on the PSNR curves by the downward spikes. The experimental results for $n = 3$ and $n = 5$ clearly indicate that, although their PSNRs are not as steady as in the VBR results, they are far better than in the $n = 1$ case in terms of image quality consistency as reflected by the lower PSNR standard deviations.

Visually, the image degradation and inconsistency for the $n = 1$ is extremely irritating. Similar observations are gathered from all three sequences so the subsequent analysis applies equally to all three. When each frame is forced to meet the same bit rate, the effects on the video sequence is devastating. The perceived image quality oscillates from very poor to acceptable. The frames which are encoded as I frames are very distinguishable as they suffer the most drastic impairment. This is to be expected because, in most cases, the bit rates for the I frames are cut to one-third of their unconstrained VBR values. The consequences are drops of up to 6.0dB for "bus" I frames, 7.4dB for "flower" I frames, and 7.2dB for "calendar" I frames. Clearly this is not an acceptable situation and for this reason, constant bit rate sequences are never generated in this manner. Rather, a much more preferable method is to average several frames together. For the rate-control method explored in this thesis, there does not seem to be a need to use a large averaging interval to achieve respectable results. An interval that spans a distance long enough to encompass a high bit rate frame (I or P frame) and several B frames seems to provide good image quality stability. The prudent choice of n would then be a multiple of M, the distance between P frames. This is the situation for $n = 3$ as very little image quality flicker is noticeable in this case and the general image quality nearly matches that of the pure VBR example. With larger values of n, long stretches of frames may be coded with markedly higher or lower bit rates than the average. This may lead to a visible oscillations in the image quality between groups of frames. In table 3.5, this is illustrated by the higher PSNR standard deviations for $n = 5$ compared to $n = 3$. Visually, the sequences using $n = 3$ and $n = 5$ are virtually identical. Both exhibit a bit of image quality oscillations and both appear slightly inferior to the VBR sequence. Ideally, the value of n would be chosen to correspond to the buffer size of the video file server and may range anywhere from a single frame to several group of pictures.

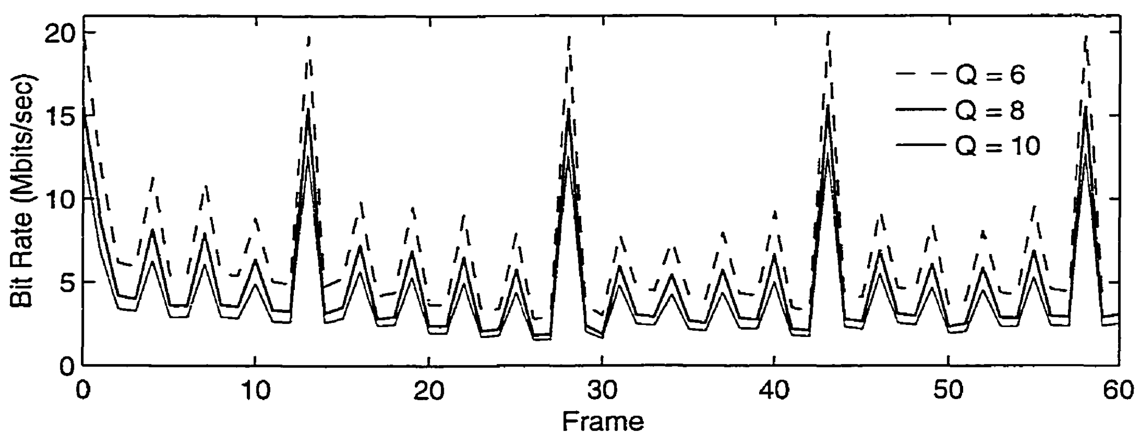
One final note regarding the subjective quality of the video sequences. The decoded bit streams represent the reconstruction of video sequences that have undergone a lossy form of compression. As such, the images have lost a certain amount of detail and sharpness compared to the original. However, the overall quality is still very presentable and can easily be regarded as 'good' quality. Impressive is the fact that the files that store the MPEG-2 compressed video information for every case discussed in this chapter (as well as for the next chapter) are all around 1Mb in size. Compared to the original sequences which each occupy over 30Mb of disk space, this represents an amazing 30:1 compression ratio. Without the MPEG-2 compression algorithm, the transmission of the sequences would require a monstrous 120Mbps channel each which is both very expensive and impractical.



(a) "bus"

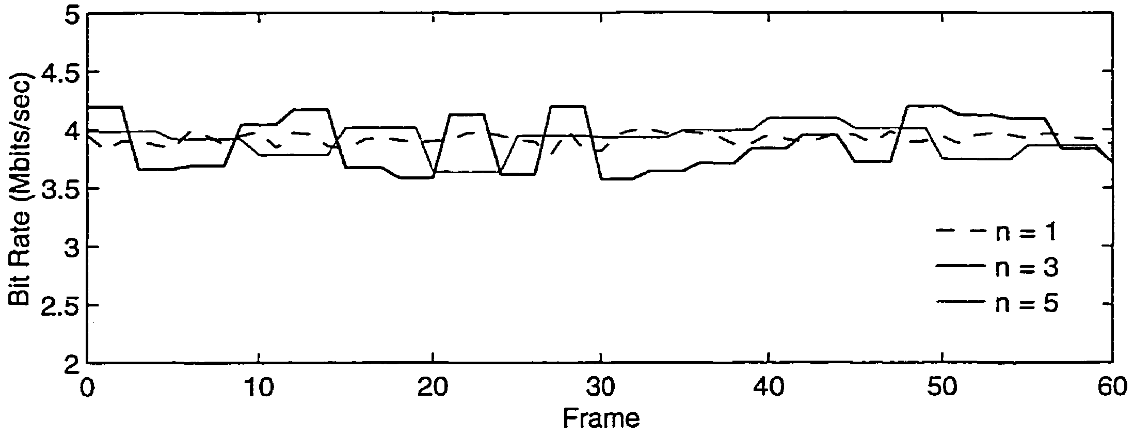


(b) "flower garden"

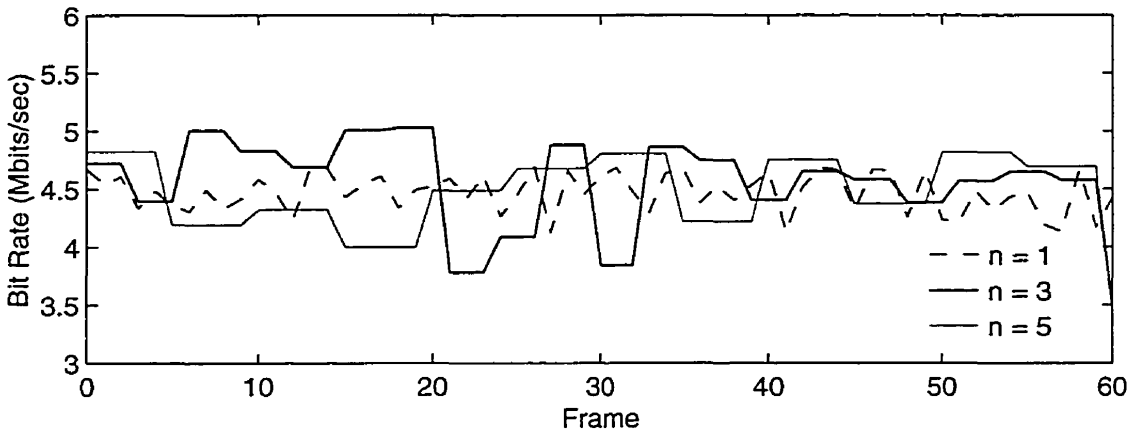


(c) "calendar"

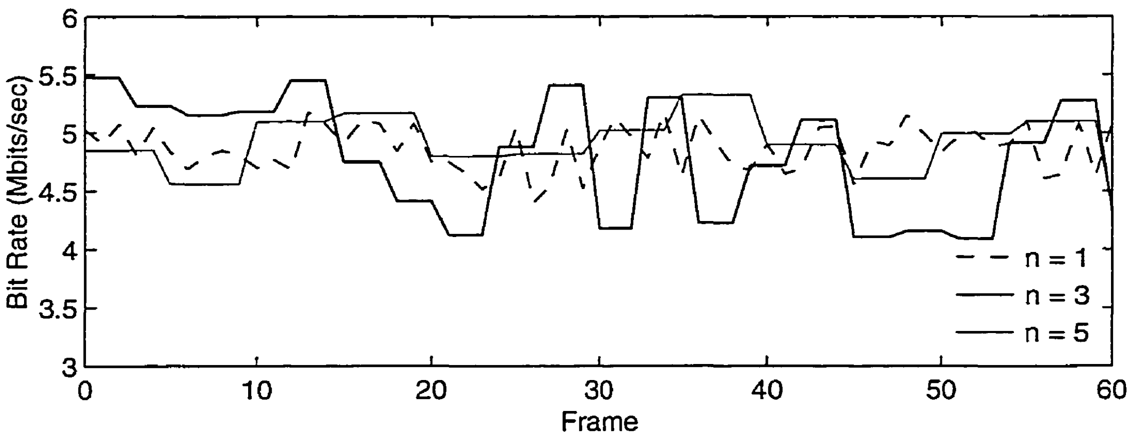
Fig. 3.4 VBR bit rate profiles for different values of Q



(a) "bus"

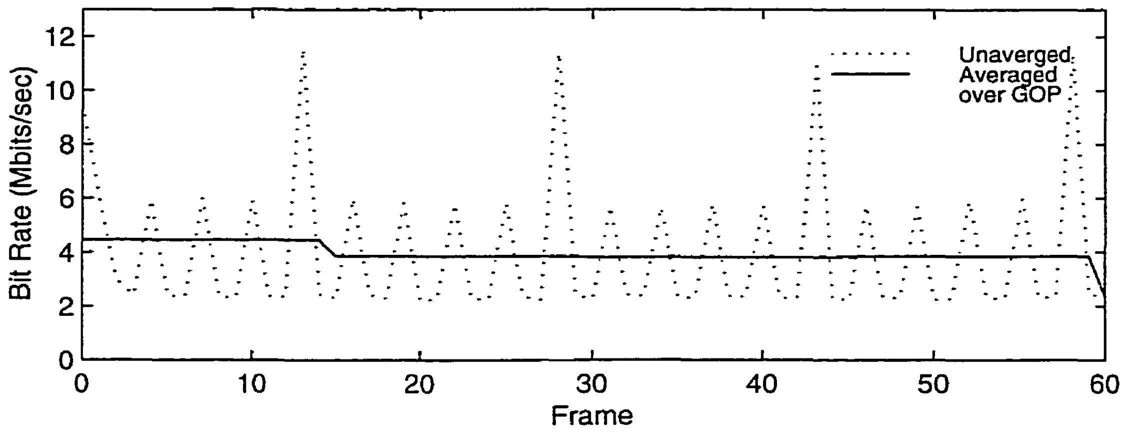


(b) "flower garden"

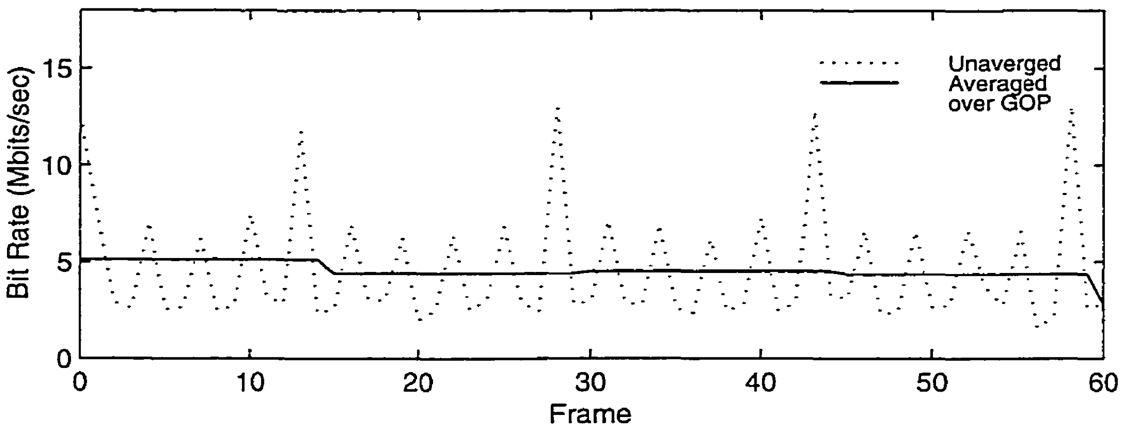


(c) "calendar"

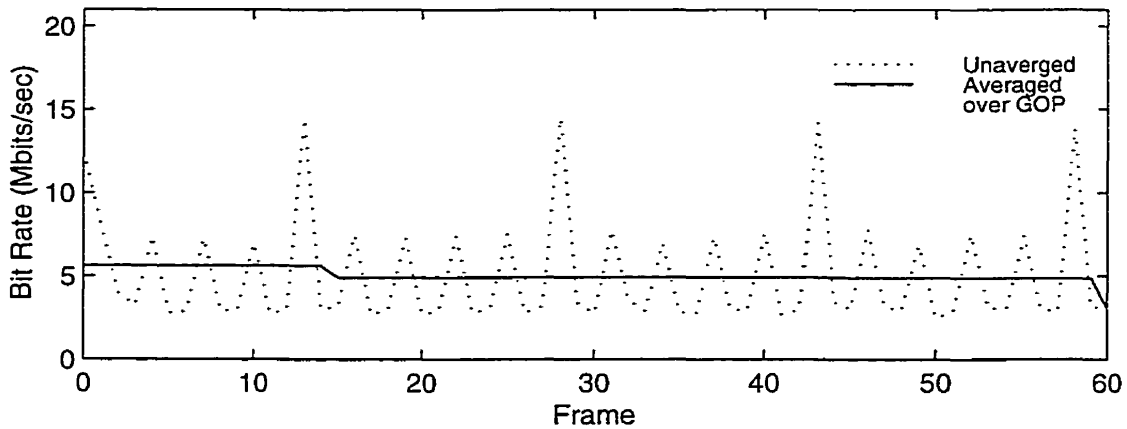
Fig. 3.5 CBR bit rate profiles for different values of n



(a) "bus"

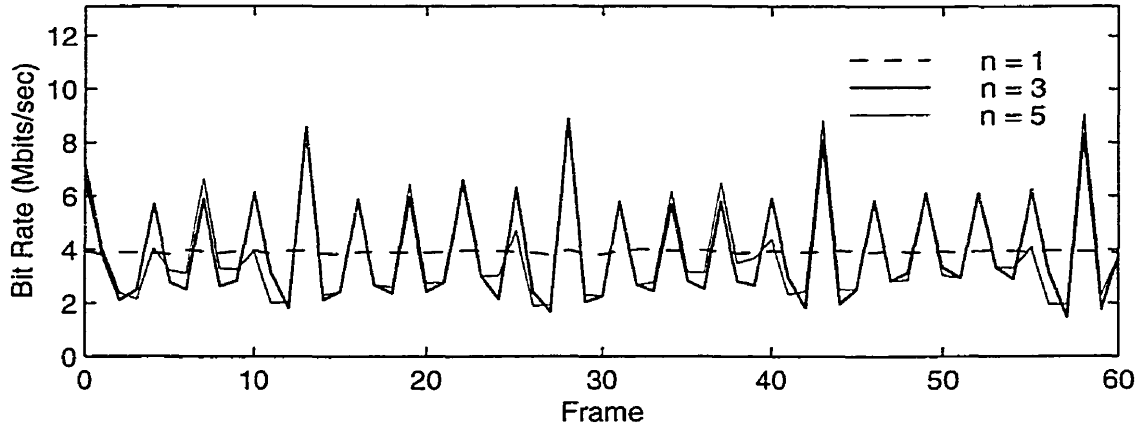


(b) "flower garden"

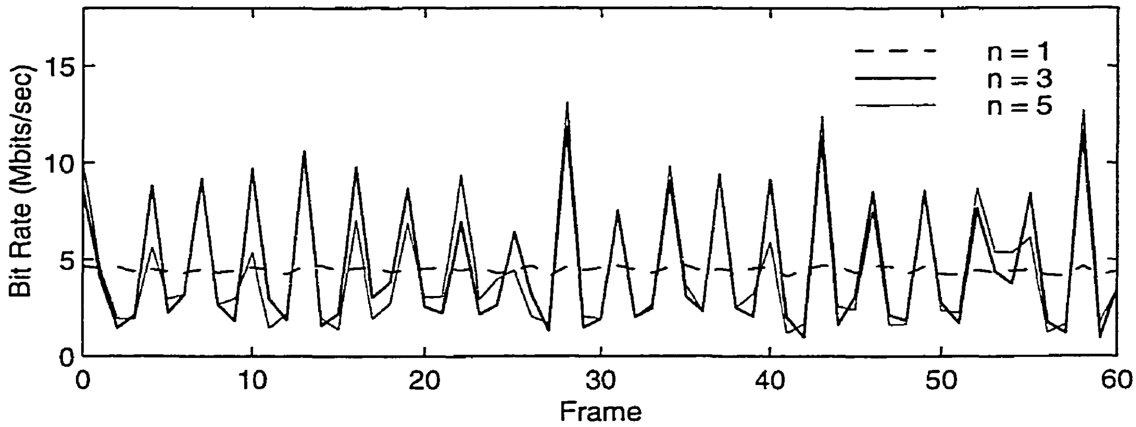


(c) "calendar"

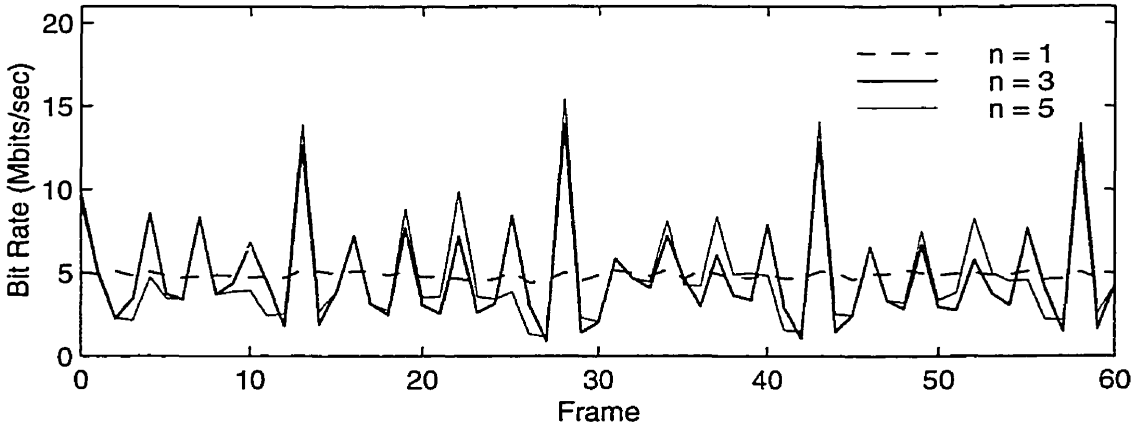
Fig. 3.6 TM5 averaged and unaveraged bit rate profiles



(a) "bus"



(b) "flower garden"



(c) "calendar"

Fig. 3.7 Unaveraged CBR bit rate profiles for different values of n

Sequence	*Q*	*max(Mbps)*	*avg(Mbps)*	*min(Mbps)*	*burstiness*	*std(Mbps)*
bus	6	12.7483	5.3195	3.2868	2.3965	2.6017
	8	10.0286	3.8482	2.2606	2.6061	2.1163
	10	8.2944	3.1358	1.9346	2.6451	1.7042
flower gar.	6	17.7821	6.1508	2.5733	2.8910	4.1803
	8	14.0688	4.4462	1.7206	3.1642	3.3546
	10	11.6143	3.6191	1.5010	3.2092	2.6825
calendar	6	20.0580	6.8829	2.7787	2.9142	4.5506
	8	15.6427	4.8884	1.8072	3.2000	2.6765
	10	12.8102	3.9246	1.5217	3.2641	2.9693

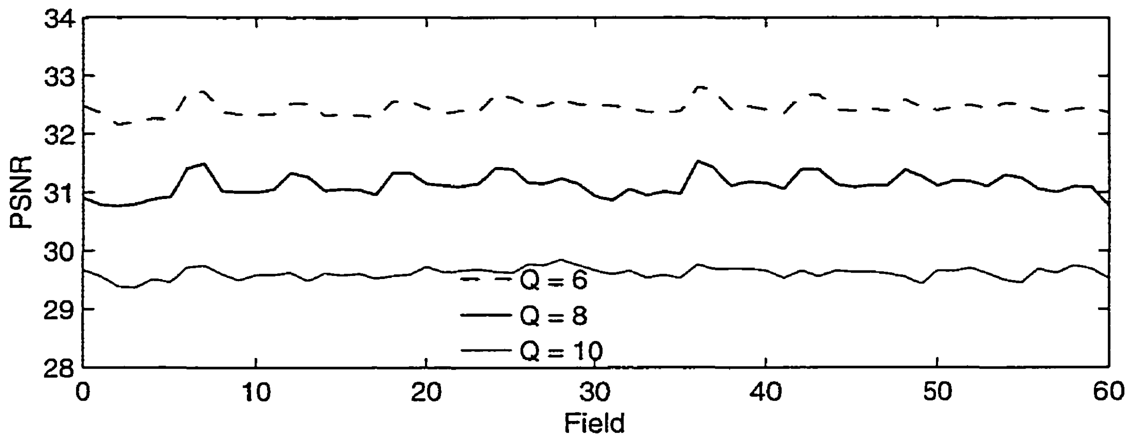
Table 3.1 VBR coded bit rate statistics

Sequence	*n*	*max(Mbps)*	*avg(Mbps)*	*min(Mbps)*	*burstiness*	*std(Mbps)*
bus	1	3.9996	3.9203	3.7838	1.0202	0.0512
(target:	3	4.1978	3.8792	3.5731	1.0821	0.2331
3.8482Mbps)	5	4.0976	3.9069	3.6342	1.0488	0.1290
flower gar.	1	4.6925	4.4698	4.1258	1.0498	0.1681
(target:	3	5.0284	4.5679	3.4877	1.1008	0.3730
4.4462Mbps)	5	4.8232	4.4977	3.4877	1.0724	0.3042
calendar	1	5.1766	4.8639	4.3994	1.0643	0.1977
(target:	3	5.4758	4.7996	4.0930	1.1409	0.5060
4.8884Mbps)	5	5.3241	4.9257	4.3154	1.0809	0.2304

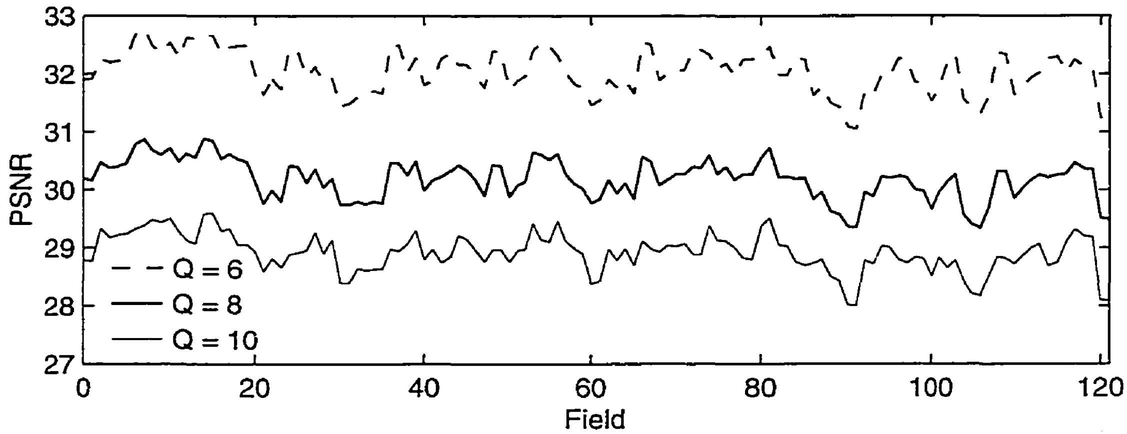
Table 3.2 CBR coded bit rate statistics

Sequence	*max(Mbps)*	*avg(Mbps)*	*min(Mbps)*	*burstiness*	*std(Mbps)*
bus (target: 3.8482Mbps)	4.4574	3.9705	2.3630	1.1226	0.3385
flower gar. (target: 4.4462Mbps)	5.1276	4.5798	2.8214	1.1196	0.3815
calendar (target: 4.8884Mbps)	5.6150	5.0416	3.0797	1.1137	0.4040

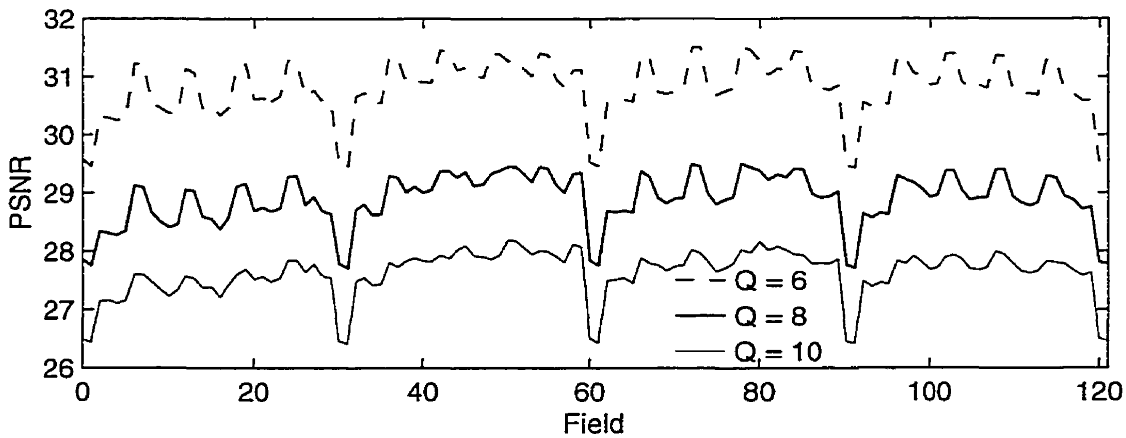
Table 3.3 TM5 coded bit rate statistics



(a) "bus"

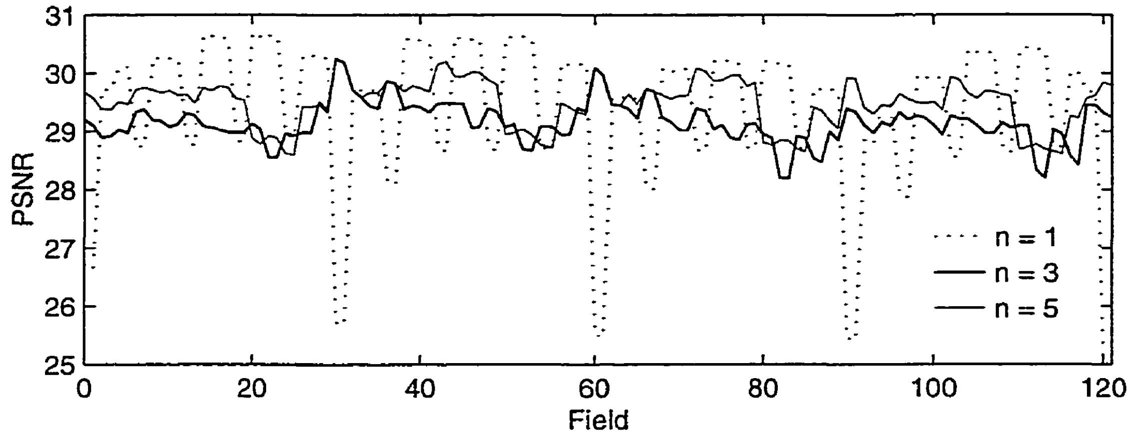


(b) "flower garden"

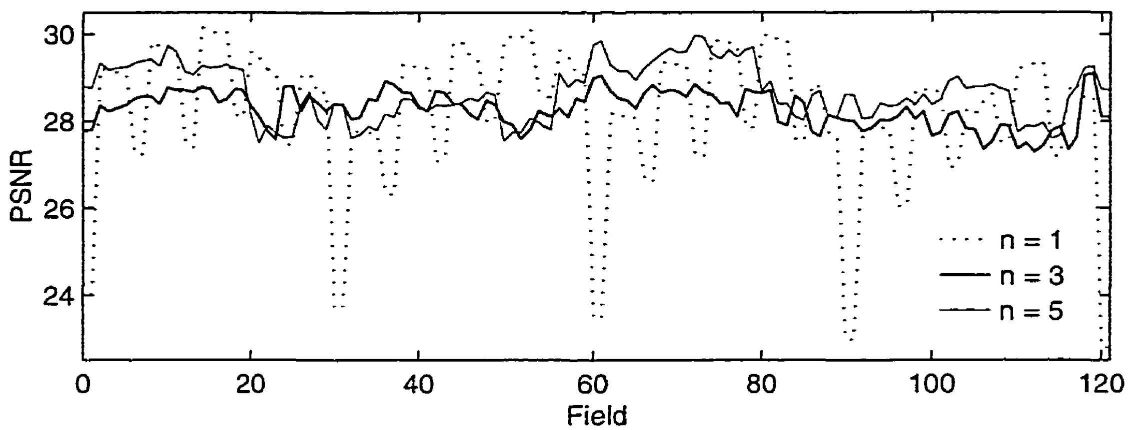


(c) "calendar"

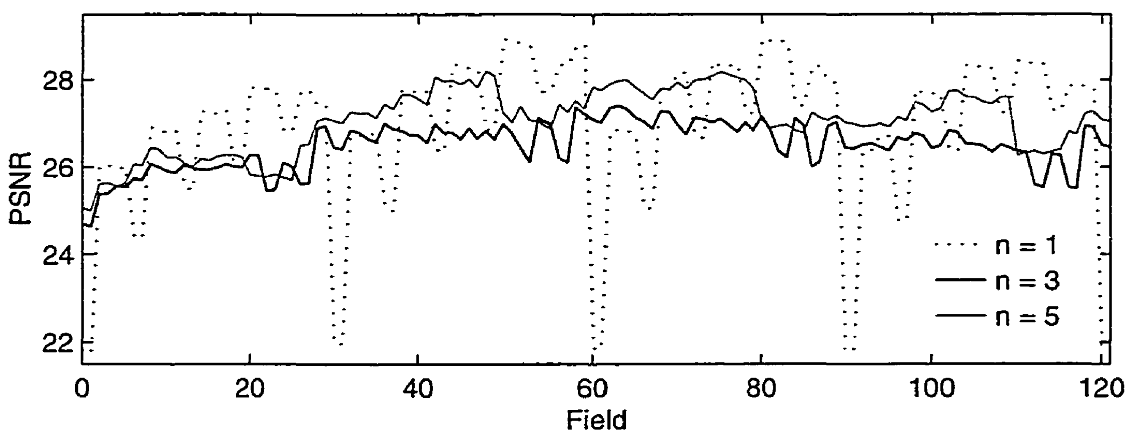
Fig. 3.8 VBR luminance PSNR curves for different values of Q



(a) "bus"

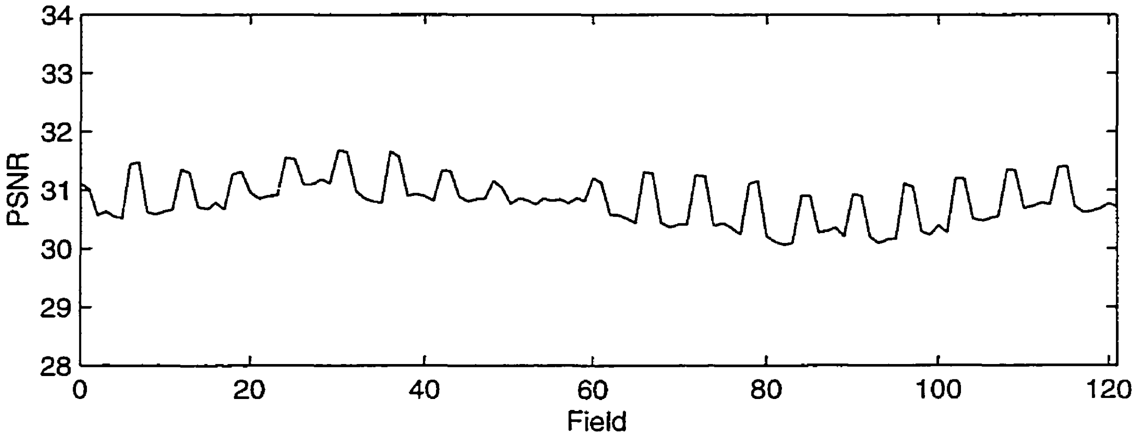


(b) "flower garden"

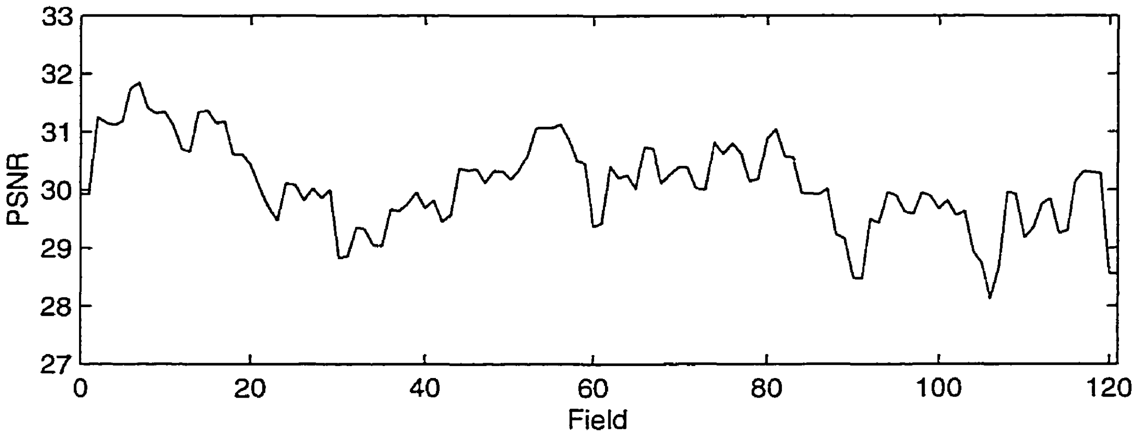


(c) "calendar"

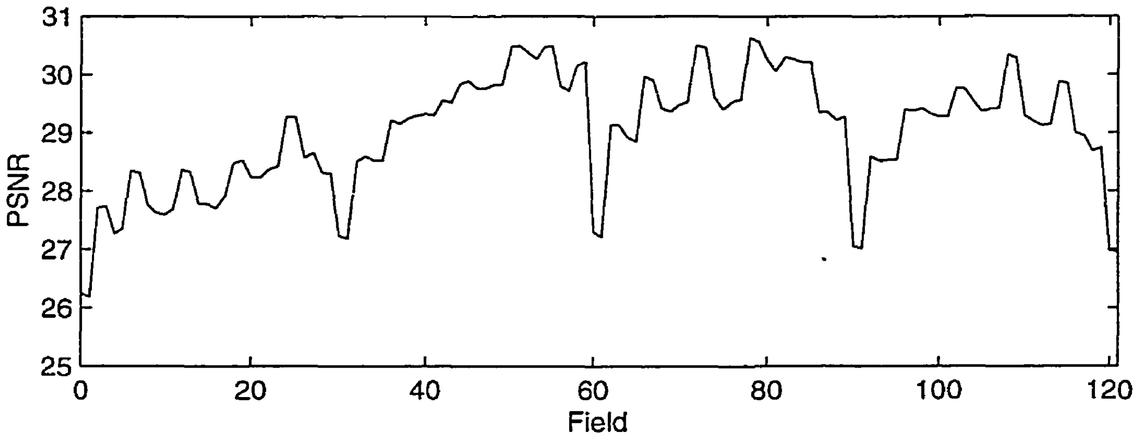
Fig. 3.9 CBR luminance PSNR curves for different values of n



(a) "bus"



(b) "flower garden"



(c) "calendar"

Fig. 3.10 TM5 luminance PSNR curve

Sequence	*Q*	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	6	32.8061	32.3745	32.0352	0.1766
	8	31.5366	31.0499	30.5028	0.2277
	10	29.8577	29.5438	29.2429	0.1221
flower gar.	6	32.7774	32.0382	31.0522	0.3725
	8	30.8816	30.1890	29.3449	0.3403
	10	29.5973	28.9186	28.0054	0.3291
calendar	6	31.5272	30.7977	29.4309	0.5142
	8	29.5063	28.8972	27.7159	0.4538
	10	28.1983	27.6216	26.4177	0.4199

Table 3.4 VBR coded luminance PSNR statistics

Sequence	*n*	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	1	30.6918	29.3658	24.9971	1.3565
	3	30.2467	29.1520	28.1943	0.3580
	5	30.2467	29.4879	28.5986	0.4140
flower gar.	1	30.1510	28.1474	22.7813	1.7438
	3	29.1037	28.2764	27.2952	0.4200
	5	29.9739	28.6737	27.4911	0.6334
calendar	1	28.9429	26.8746	21.7225	1.8257
	3	27.4131	26.4813	24.6473	0.5610
	5	28.1995	27.0514	25.0065	0.7508

Table 3.5 CBR coded luminance PSNR statistics

Sequence	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	31.6857	30.8224	30.0634	0.3925
flower gar.	31.8472	30.0915	28.1267	0.7548
calendar	30.6288	29.0154	26.1907	1.0152

Table 3.6 TM5 coded luminance PSNR statistics

Chapter 4

Bit Stream Traffic Shaping

In actuality, an unconstrained variable-bit-rate coder has very limited practicality. This is attributed to the restrictions on the end-to-end delay being strict enough that re-transmissions for data packet loss are not possible during transmissions to the client. Furthermore, any delays greater than a certain tolerance may result is the received packet arriving too late to be processed — this too may be considered a network-induced error. Thus, the importance of traffic shaping MPEG-2 bit streams becomes apparent. By employing a *rate controller* in the source coder, the output data stream can be conditioned to conform to the expectations of the network. In this manner, the amount of network-induced information loss can be anticipated to be negligible. This is an important consideration in packet video.

4.1 Motivation

Bit errors attributed to the electronics and optics of the transmission medium are obviously beyond the control or predictability of the source (see Karlsson [20, 21] for a discussion of network induced errors). For these errors, several methods to limit or mask the losses are available. These include, loss concealment, forward error correction, and layered coding. However, for modern networks, especially those using fiber optics technology such as a Synchronous Optical Network (SONET), the bit-error rate is extremely low, even to the point where it is no longer a consideration. A more serious source of information loss is the disposal of packets by the network itself to preserve its integrity. For an ATM network, the self-protection mechanism is a policing function based on the Leaky Bucket Algorithm. Since the behavior of the network's policing function is completely understood a priori to the transmission, a source coder should be able to take preventative measures to ensure that the instances of network interference are minimal.

Network interventions during the transmission of MPEG-2 data packets are especially detrimental because bucket overflows will almost certainly occur during the transmission of intra-coded frames. With the rate controller imitating the functionality of the network policer, an MPEG coder would be able to accurately predetermine the instances at which its output would trigger a corrective response from the network and then employ traffic shaping. Supplemented by the fact that the encoding is performed completely off-line (meaning that real-time encoding and transmission is not expected), the coder has the opportunity to recode segments of data which it foresees as violating the traffic parameters. In this manner, a bit stream that will completely conform to the traffic descriptors can be created and thereby provide strong assurance that the transmission will reach the destination without interruption or network intervention, according to the set-up quality of service contract. Logically, for the case of ATM networks, the MPEG-2 coder should incorporate a form of the Leaky Bucket algorithm into its rate-control function to monitor the output bit stream and provide feedback information to the quantizer. Essentially, from the point of view of the encoder, the network is a black box whose only relevant parameters to consider are the traffic descriptors (peak bit rate, sustained bit rate, burst length).

Another potential advantage of traffic shaping is that if the video is coded to meet the strict constraints of the network traffic descriptors, then additional traffic shaping at the network interface of the file server no longer becomes necessary. A server will usually administer its own traffic shaping by buffering several frames of coded video and then decreasing the service rate as needed. However, this has the implication that additional, and possibly error-inducing delay is introduced. Video files that have been traffic shaped beforehand may be transmitted strictly at their viewing rate, meaning that, for instance, a frame rate of thirty frames per second would result in the data for each frame being transmitted entirely within one-thirtieth of a second.

In the alternate situation where the video is encoded and transmitted in real time, a close interaction between the coder and network may be needed. In the event that the network offers a set of traffic parameters that are more restrictive than the ones used to shape the video bit stream, an avenue still remains by which network conformance is still attainable. By employing a *dynamic rate shaper* as an intermediary between the encoder and the network, the bit stream can be modified in real-time to meet the bit-rate characteristics expected by the network. Such schemes are outlined by Reininger et al. [12], Eleftheriadis and Anastassiou [30], and Zhang and Knightly [31].

The remainder of this chapter is devoted to describing the operation of the rate-controlling features incorporated into the VBR coder, and then, through experimental evidence, to illustrate the value of rate-constrained coding over both constant-bit-rate and variable-bit-rate services.

4.2 The Rate-Constrained Coder

The enhanced, rate-constrained coder uses the ATM Forum's specification of the Leaky Bucket traffic-monitoring algorithm as the primary component of its traffic-shaping function. However, considering the off-line operation of the software coder, special considerations and modifications have to be made to fit the real-time algorithm of the Leaky Bucket into the rate controller.

4.2.1 The Traffic Shaper

Figure 4.1 provides an illustration of the overall layout of the coder with the rate controlling features. The components of the block diagram enclosed by the dotted line collectively comprise the traffic shaper and would reside inside the feedback block of figure 2.11 on page 23.

Since it operates in a variable-bit-rate manner, the MPEG-2 coder is free to generate as many bits as needed to compress each input frame. Initially, the quantization scale is assigned a value, Q_{init}, which also serves as the lower bound for Q. A lower bound is needed because the purpose of traffic shaping is to momentarily limit the data flow, not to exploit as much available bandwidth as possible. Furthermore, network operators would be very reluctant to provided more resources, and hence, better quality, than agreed upon in the connection contract. The values of K_P and K_B are

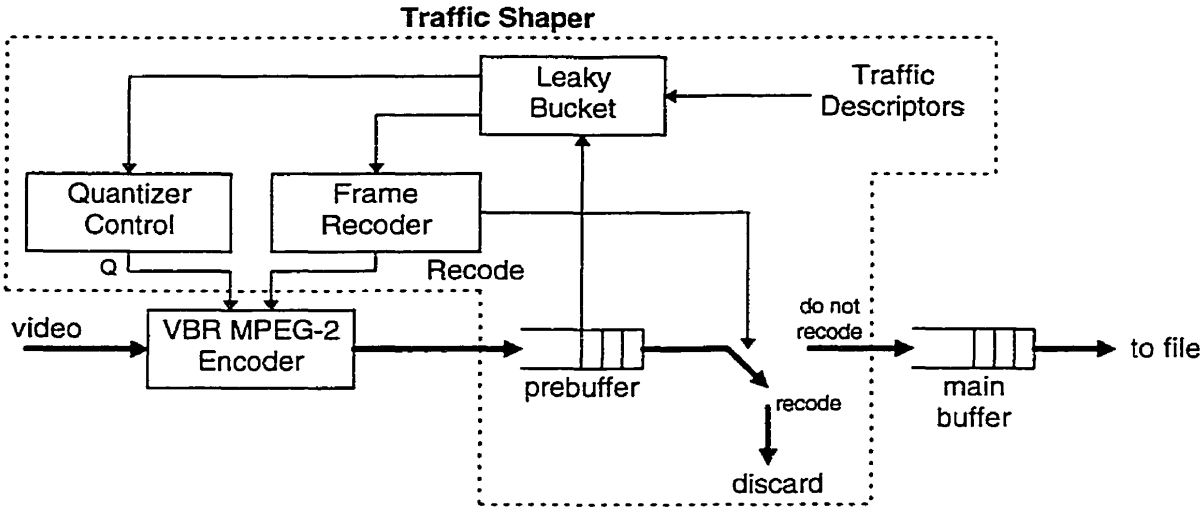


Fig. 4.1 Block diagram of the rate constrained coder

fixed before the onset of the encoding process. In general, they are assigned the values used in the open-loop encoder to promote constant image quality.

The decision process of the traffic shaper is summarized by the flowchart in figure 4.2. Before entering the main buffer, compressed video data is redirected and temporarily held in a prebuffer. Whether or not the data in the prebuffer is allowed to pass onto the main buffer depends upon the its conformance to the traffic descriptors as judged by a pair of Leaky Buckets. If the data is deemed acceptable then the bits are released to the main buffer where the information is eventually written to a file. Otherwise, one or more of the traffic descriptors has been violated and the prebuffered data is discarded while the coder is reset to recode the discarded frames at a lower bit rate (higher quantization scale factor). For this purpose, the values of certain internal parameters used by the coder during the compression of the frames are saved prior to processing the frames; in the event of a *recode* decision, theses parameters are re-instated to effectively "rewind" the encoder to the beginning of the n frames. The flow chart shows that two Leaky Buckets are employed in the decision making process. This is because one is needed to monitor the peak rate and the other to monitor the sustained rate and burst length.

Three key components comprise the traffic shaper. They are the *Leaky Bucket*, the *quantizer control*, and the *prebuffer and frame recoder*. Each is now discussed

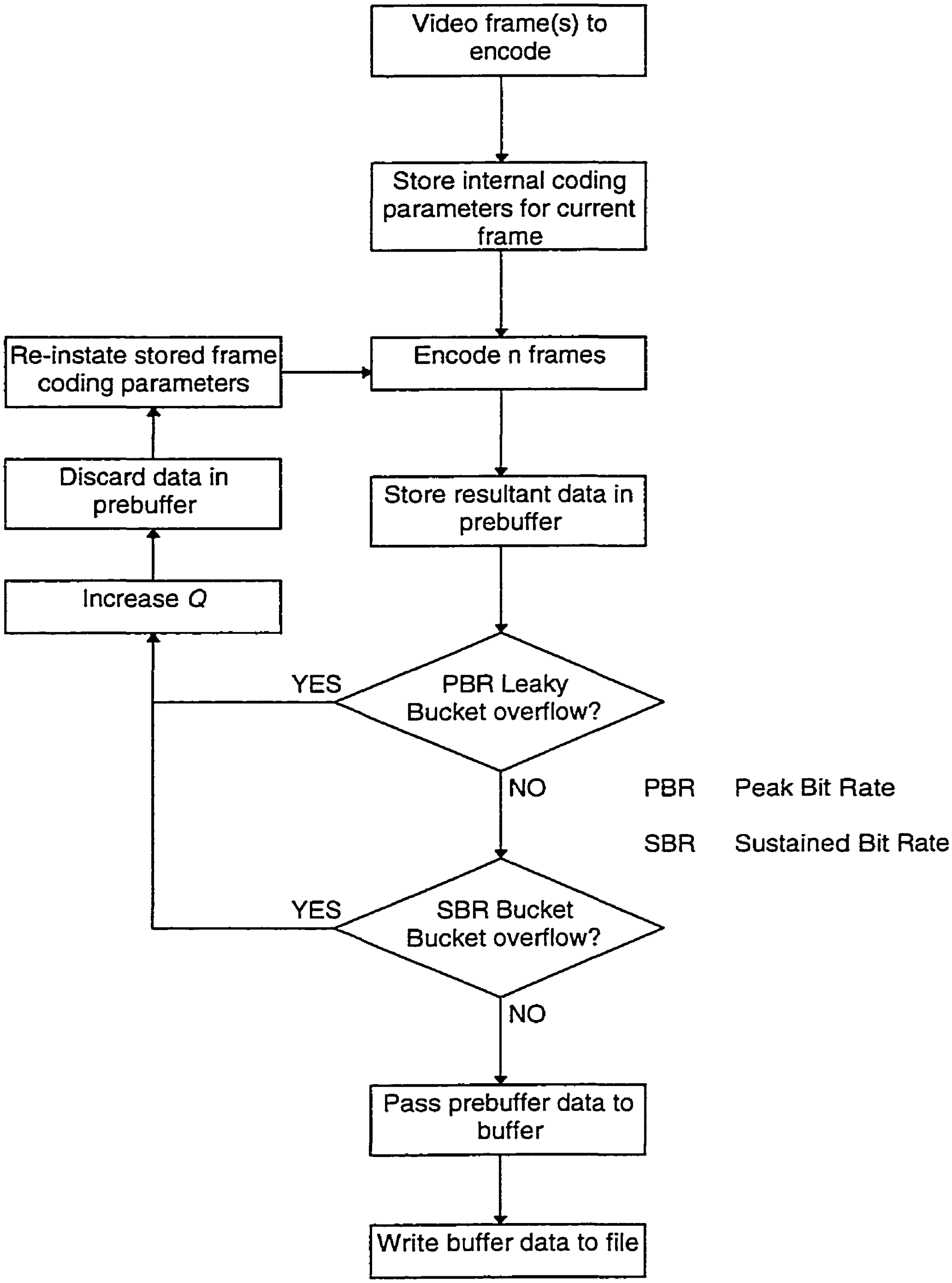


Fig. 4.2 Flowchart of the traffic shaping function

individually in detail.

4.2.2 The Modified Leaky Bucket

Although the Leaky Bucket system employed in the rate-constrained VBR MPEG-2 coder is based on the ATM Forum description of the algorithm (section 2.3.2) there are a couple of important distinctions. The obvious difference is that the ATM Forum's specification deals with packets while in this study, bits are being considered. This is not a concern because the substitution the bit arrival time for the ATM cell arrival time in the description of the algorithm does not alter the Leaky Bucket's functionality. While the data rates required by the network are in terms of cells, the constant packet size used in ATM suggests a straightforward conversion from bit rate to cell rate. As a result, the desired peak and sustained cell rates can be replaced with the peak and sustained bit rates (R_P and R_S). What is a concern, however, is that the coding is performed off-line while the Leaky Bucket requires information about the interval times between consecutive data arrivals. When coding video through an off-line software program, the only notion of actual time that would have elapsed in the video sequence is the frame period, $1/frate$, where *frate* is the display frame rate. This leads to a major, but usually correct, assumption that the frame data for any particular frame is sent collectively rather than in separate transmissions. This is a valid supposition as many video-file-server systems are based on transmitting portions of data representing a predefined length of video at a time (for example, Neufeld et al. [15]). Under this assumption, the inter-bit arrival time for every bit in a given frame will be equal. This, in turn, implies that the Leaky Bucket need only work with one inter-bit arrival time per frame (or group of n frames if $n > 1$) instead of considering every generated bit. In essence, the Leaky Bucket is considering the n frames as a single arrival and continuing with this reasoning, the maximum burst size (mbs) should specify the duration at which the peak bit rate can be sustained in terms of number of frames instead of number of bits. The modified GCRA adopted by the traffic shaper is summarized in figure 4.3.

With consideration to the differences noted above, the translation of the inter-arrival time, T, into additional bucket occupancy and the calculation of bucket leak rate, I, are carried out exactly as outlined for the ATM GCRA. The equations for

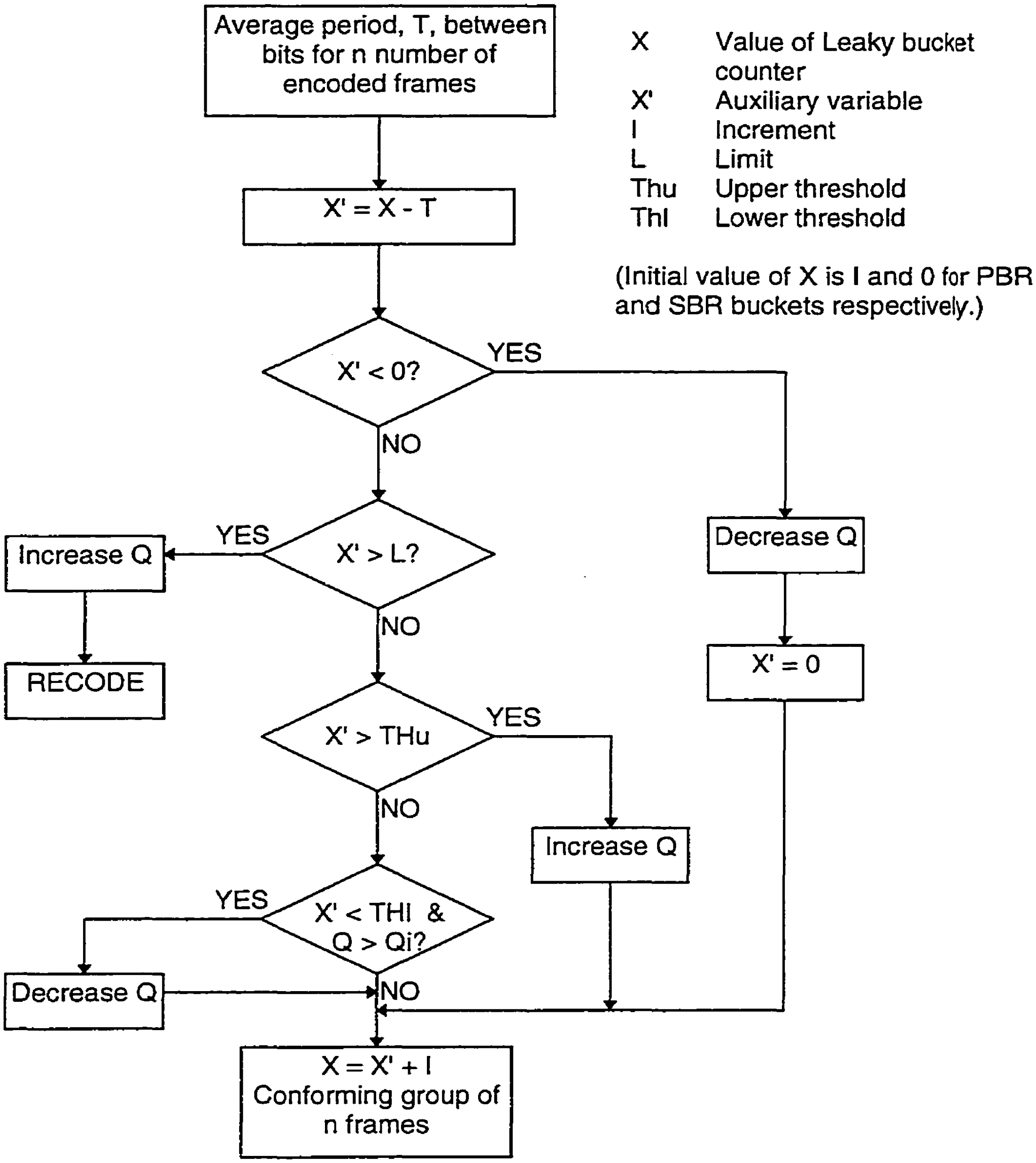


Fig. 4.3 Modified Leaky Bucket algorithm used in the rate controller

the leak rates remain

$$T_P = 1/R_P \tag{4.1}$$

$$T_S = 1/R_S \tag{4.2}$$

where the subscripts P and S refer to the peak-bit-rate (PBR) and sustained-bit-rate (SBR) Leaky Buckets respectively. However, the size of the SBR bucket, τ_S, has one minor modification. In place of equation 2.4, the bucket size is determined by

$$\tau_S = (mbs/n - 1)(T_S - T_P) \tag{4.3}$$

The scaling of the mbs by $1/n$ stems from the fact that a stretch of n frames is treated as just one entry by the rate controller. For example, if $n = 2$ and $mbs = 4$, then the rate controller would evaluate two frames at a time. Therefore, after four frames have elapsed, the rate controller will have actually only processed two inter-arrival times. Thus, from the point of view of the rate controller, only two frames have been evaluated.

While the bucket level for the SBR bucket may be initialized to zero, the same is not true for the peak-bit-rate bucket. Recall that a PBR bucket will always have a bucket size of zero. Accordingly, the observed traffic will only conform if the average bit inter-arrival time is greater than T_P, or equivalently, $X' \leq 0$ (see figure 4.3). However, if the level counter, X is initialized to zero, then the first segment of traffic being policed will always conform, regardless of the bit rate, because the first value of X' will always be negative. The remedy for this is simply to initialize the PBR value of X to T_P.

Whereas the ATM Forum's GCRA tags offensive cells to be discarded, the Leaky Bucket used in the traffic shaper instructs the coder's quantizer to increase or decrease the coarseness of the coefficient quantization, and in the extreme case, recode the entire group of n pictures. Upper and lower thresholds are included in the algorithm in case they may be helpful in the avoidance of bucket overflows or starvations.

Thresholds

Although an upper threshold is included in the policing algorithm to help reduce the occurrences of bucket overflows, its usefulness is not entirely obvious. The SBR bucket may benefit from an upper threshold as discussed in 4.4.2. For monitoring peak rates, the thresholds are completely useless because the PBR bucket always possesses a zero capacity, and thus, the arrival time for a sequence of frames can only either overflow or underflow the bucket.

The lower threshold should be set to a level so that an empty bucket is avoided. An empty bucket signifies that either the bit rate is being overly-suppressed, or that the defined sustained rate overestimates the actual average bit rate. Assuming that the former is true, the quantizer scaling factor is being over-zealously increased above the initial value and is therefore causing the generated bit stream to underutilize the bandwidth allocated to it. This has the consequence that the decoded sequence will playback with a poorer picture quality than expected. Using the strategy that the SBR bucket should be at least half full, the lower threshold, *THl*, is set to half of the bucket size. If the bucket level decreases beyond half capacity, the rate controller will increase the bit rate of the next n frames to increase the bucket occupancy.

4.2.3 The Quantizer Control

The function of the quantizer control is to increase or decrease the bit rate through adjustments to the quantization scaling factor upon the request of the Leaky Bucket. The determination of the extent with which to adjust Q then becomes the central problem. The simplest and easiest solution is to increment or decrement Q by a constant value. This method sacrifices some efficiency as the most suitable value of Q is found through a linear search. Nonetheless, this formula has proved completely adequate for the rate-constrained coder described here. Another possibility is to use a multiplicative constant instead of an additive one. However, the obvious deficiency with this approach is that, depending upon the constant, at higher values of Q, the change in value may be very large, while at lower values, Q may not change at all (remember that Q is restricted to an integer). Furthermore, depending upon the values for K_P and K_B, the P and B frames may have their bit rates increased or decreased more than desired with a single step change to Q.

A more sophisticated solution is to use an encoding statistic known as the *Global Complexity Measure* to predict the quantization scaling factor for the upcoming frames [11]. The predicted value of Q can then be used as the starting point for the next series of frames, thereby, reducing the effort needed by the coder to find the correct Q to produce the needed bit rate.

Constant Bit Rate Mode

When the coder is operating in constant-bit-rate mode, a slightly different quantizer control is used in an effort to maintain the consistency of the bit rate. Although altering the quantization parameter in small increments would serve to produce less disturbing image quality changes between frames, it does not result in the greatest effort to maintain a fixed bit rate. For this reason, when operating in CBR mode, the coder will always jump back to the initial value of Q, Q_{init}, after successfully coding a segment of video. This ensures that the bit rate will not be too low to fully utilize the allocated bandwidth.

4.2.4 The Prebuffer and the Frame Recoder

Collectively, the prebuffer and the frame recoder are responsible for ensuring the proper recoding of the frames that have been rejected by the Leaky Bucket. A prebuffer is employed rather than strictly using the main buffer for the issue of simplicity. The main buffer has associated with it, many complex functionalities linked to the original TM5-based rate controller. Trying to disconnect all the links would have been rather time consuming and treacherous; it proved far simpler to just insert an additional buffer. The prebuffer is given a size equivalent to the maximum number of bits allowable under the declared traffic descriptors. This is equal to the product of n and the maximum number of bits allowable per frame

Resetting the MPEG-2 coder to recode a segment of the video sequence requires careful attention. Several internal parameters must necessarily be reset to their exact values at the onset of coding the most recent group of frames or else the frames will not be processed properly. The frame recoder accomplishes this by duplicating these parameters in an area of memory prior to the coding of the n frames. After a recode order is issued by the Leaky Bucket, values of these stored parameters are immediately replaced into the coder. This process can be loosely described metaphorically as giving the coder "selective amnesia" of the last n frames. The parameters duplicated by the recoder are mostly variables that keep track of the coder's position in the video sequence. This includes the frame counter and counters for the forward and backward motion vectors. At the time these parameters are stored, they must be accompanied by the complete image data for the current forward and backward reference frames. This is essential for the any P and B frames that may need to be recoded; recall that motion compensated frames will code and transmit the spatial error based upon the image interpolated from the reference frames. The precise structure defined in the coder to store these parameter is provided in figure 4.4.

4.3 Limitations

The outlined process by which a frame, or group of frames, are recoded to maintain conformance to the traffic descriptors is not always practical. If the percentage of frames that need to be recoded is large, as is normally the case when trying to produce constant-rate-bit streams, the amount of processing time can increase by six or seven fold compared to the open-loop cases. Furthermore, the lack of a sophisticated quanitzer control aggravates this situation. For the purposes of this research, this is acceptable because the CBR sequences are generated only for comparison. In any case, as long as the traffic descriptors give a reasonable approximation to the unconstrained VBR bit-rate statistics (in particular, the sustained bit rate), the recoding will have a minimal effect upon coding process time and complexity as only I frames will usually need to be reevaluated.

4.4 Experiments and Results

The experiments conducted in this research make the assumption that the values of the traffic descriptors that will be negotiated with the network are known beforehand, or at least their upper limits are already established.

Using the same three sequences considered in the previous chapter, an examination of the influence each individual traffic descriptor has upon the resultant bit stream is

```
typedef struct {
  int next,               /* next reference frame number */
      prev,               /* previous reference frame number */
      d_f,                /* forward motion field index */
      d_b,                /* backward motion field index */
      n_gop_cod,          /* Group Of Picture counter */
      max_pos,            /* maximum temporal position */
      t_pos,              /* temporal position */
      fcount,             /* encoded frame count */
      totbits;            /* total coding bits count */
  long tbits;             /* total coding bits for n frames */
  int b;                  /* B frame number              */
                          /* note: b<0 --> frame is I or P */
  unsigned char **nxt_im; /* spatial content of next ref. frame */
  unsigned char **prv_im; /* spatial content of prev. ref. frame */
  int Ni, Np, Nb;         /* bit counts for I, P, and B frames */
  BUCKET pbr, sbr;        /* PBR and SBR Leaky Bucket information */
  PICTURE pict;           /* input picture information */
  int status;             /* param. used for  reading dspl files */
  DSPL_INF dinfo;         /* displacement vector information */
  IMAGE image;            /* current picture information */
  int mb_size[2];         /* macroblock structure information*/
} PRIOR;                 /*** struct storing prior values of vars */
                         /*** in the event of a RECODE          */
```

Fig. 4.4 Declaration of the structure used to record the coding parameters now presented. The case of $mbs = 1$ is never investigated because this would cause the sustained-bit-rate bucket capacity to become zero and the SBR bucket would effectively be used as a peak-bit-rate bucket, except with a lower rate. A quantization scaling factor of eight is used as the starting value for each experiment in order to facilitate a fair comparison with VBR and CBR results. Likewise, the quantization constants, $K_P = 1.45$ and $K_B = 1.81$, are again used. The value of n is always set to one, meaning that each frame is considered alone, to allow the effects of altering each traffic parameter to be more pronounced; if the bit rate is averaged over several frames, it will be better behaved and therefore, the possibility of rate-controller intervention will diminish. The threshold values used in the coding process throughout the ensuing results all use an upper threshold set to 1.0 (i.e., no upper threshold), and a lower threshold value of 0.5. The luminance data will be the focus of the following analysis and the chrominance information is reserved for the appendices. The same software and hardware setup outlined in the previous chapter is adopted here.

4.4.1 Limiting the Peak Bit Rate

Perhaps the most common use of traffic shaping is to limit the peak bit rate of the transmission. In this aspect of rate constraining, using the Leaky Bucket system to guide the rate controller produces superb results. In order to describe the effect of peak-bit-rate limiting, the three video sequences are coded using values of R_S obtained from the bit-rate statistics of the pure VBR bit streams while the values for R_P are varied; the value for mbs is always held constant at five. Figure 4.5 displays the bit rate profiles for the three sequences, each with three different values of limiting peak rate. These chosen limiting values will serve to bound the peak rates for the I frames only; using lower values would lack practicality since P frames would also start being bounded and the image distortion would be considerable. Besides the diminished bit-rate amplitudes of the I frames, the plots do not deviate far from the open-loop VBR case. Predictably, the instances with the lowest bounding peak rate for each sequence experience the greatest difficulty in following the open-loop trace. In order to reduce the number of encoding bits for each I frame to such a low level, Q must be increased considerably from its initial value. Because of the mandate of graceful image quality restoration, the next frame or frames will still continue to suffer an underallocation of encoding bits. This explains the existence of the troughs after the I frames for the lowest values of R_P. Another peculiarity that can be expected from limiting the peak bit rate is the elevation of the bit rates for some of the motion-compensated frames immediately following an I picture. The reason for this is the higher quantization error present in the reference I frames. As long as Q is not too low, the additional error translates to an increased amount of error signal that must be coded and packaged with each P or B frame. The table containing the numerical bit rate information is located immediately following the graphs. To re-iterate, R_S only defines an upper limit for the average bit rate and the rate controller makes no effort to approach this value. Therefore, it is not surprising that as the value of R_P decreases, so does the observed average bit rate.

Quantitatively and qualitatively, the increasing confinement of the allowable peak bit rate seems to have little or no real impact upon the average PSNR. Figure 4.6 and table 4.2 provide the related information. The calculated data indicates that each decrease of 1Mbps in R_P typically sacrifices less than 0.2dB in terms of average PSNR — an imperceptible quality loss. However, the major consequence of lowering R_P is the sharp rise in the level of image inconsistency. The sharp dips in the PNSR curves for all three sequences give a clear indication of this. Likewise, the PSNR standard deviations are notably higher than the values obtained for the pure variable-bit-rate results. Upon viewing the video sequences, the fluctuating image qualities are found to be detectable and range from very minor to annoying, depending upon the traffic descriptors chosen. For instance, in the "bus" sequence, only a very slight image quality drop, compared to the unconstrained case, is perceived and only in the low detail areas (such as the image of the actual bus) when R_P is set to 8Mbps. As the peak bit rate decreases the amount of distortion becomes increasingly visible as well as periodic because the majority of bit reduction occurs at or around the I frames. For $R_P = 7$Mbps, the distortion is not visible enough to be considered disturbing but for $R_P = 6$Mbps, it is close. Similar conclusions can be derived from viewing the other two sequences. However, an important consideration is the percentage of decrease in the peak bit rate. For "bus," the R_P is 20.2%, 30.2%, and 41.2% below the unconstrained-peak-bit rate for $R_P = 8, 7,$ and 6Mbps respectively, while for "flower garden", the percentages are 14.7%, 28.9%, and 43.1% and for "calendar", the percentages are

23.3%, 36.1%, and 48.9% when $R_P = 12, 10$, and 8Mbps. As a result, "flower garden" suffers the least amount of impairment of all the experiments when its peak is limited to 12Mbps while the sequence "calendar" experiences, relatively, the most noticeable image quality degradation. When the peak bit rate needs to be restricted to such a low value, a better solution would be to recode the sequence altogether.

In terms of pure statistics, the traffic-shaped sequences generally still outperform the constant-bit-rate examples and, as long as the peak is not limited too excessively (around 20% below the unconstrained value), the results are comparable to the Test Model 5 generated results despite the fact that a lower bit rate is used.

4.4.2 Limiting the Sustained Bit Rate

Unfortunately, the rate controller does not seem to enjoy as much success in limiting the sustained bit rate as it does with enforcing the peak bit rate. The very long reaction time of the Leaky Bucket can be blamed for this weakness. In order for the algorithm to realize that the average rate is being violating, its bucket level must necessarily exceed its capacity. However, under most normal sets of values for the traffic descriptors, the bucket generally fills up slowly and a significant number of frames may have to pass before the capacity is reached. The capacity of the SBR bucket is directly proportional to the value of mbs and, therefore, adjusting this parameter provides a method to increase the sensitivity of the Leaky Bucket. This is evident for "bus" by inspecting figure 4.7(a). The curves in the graph are differentiated by dissimilar values of the maximum burst size which progress from $mbs = 2$ to $mbs = 5$. At the extremes, $mbs = 5$ incurs very little traffic shaping intervention (compare figure 4.7(a) and table 4.3 with figure 3.4(a) and table 3.1) while for $mbs = 2$, the bit rate shows signs of the rate controller imposing restrictions after the first few frames. The other intermediary cases suffer similar cutbacks but at later frames. These different times until onset of the rate-controller intervention is easily explained by considering the size of the SBR buckets used to police each situation, i.e., the smaller the value of mbs, and hence the bucket capacity, the sooner the bucket will overflow. Once the bucket capacity has been reached and exceeded, the rate controller will recode the violating frame until the bit rate is low enough that the bucket will not overflow. A problem arises if the current frame that causes the overflow is an I frame. In likelihood, the quantization factor will have to be reduced considerably. In addition, the next several frames will likewise have their bit rates and image qualities deflated and will only see a slow increase in both because of the efforts of the coder to instill a gradual quality change. This analysis is supported by the PSNR plots in figure 4.8(a). Visually, towards the end of the sequences (except for $mbs = 5$), each exhibits, or starts to exhibit, poor quality pictures. For $mbs = 2$ and 3, the blocking effects eventually become severe enough that object boundaries in the pictures become very distorted. Towards the end of the sequence for $mbs = 4$, the visual quality level begins to fall and it can be fully expected that, given enough frames, the $mbs = 5$ coded sequence will experience similar problems. The luminance PSNR data is given in table 4.4.

For the "flower" sequence, very little traffic shaping takes place as the corresponding figure and table attests. The reason for this is that the value of R_S used is only 10.0% below the unconstrained average bit rate while for "bus", it is 22.0% lower. Except for a minor occurrence of bit rate reduction for the $mbs = 2$ case, the encoded sequences are identical to the open-loop encoded sequences. Obviously, the sequence is either not long enough to observe any rate-controller action, or the leak rate of the SBR bucket is sufficiently close enough to the average bit rate of the unconstrained sequence that no intervention is warranted.

The results of "calendar" can be considered intermediary to the results of the other two video clips. This may be expected since the percentage decrease of R_S compared to the unconstrained average bit rate is 18.2% which lies in between that of "bus" and "flower." Only when $mbs = 2$ and 3 does any traffic shaping occur. For larger values of mbs, the bucket size seems to be large enough to absorb the surge in the bucket volume around the second I frame (the second high peak of figure 4.7(c)), unlike the examples that use smaller bucket capacities which experience overflows. A strong possibility for the occurrence of the overflow at this location may be the high CBR bucket level prior to encoding this I frame. The first GOP has a noticeably higher average local bit rate than the remaining GOPs (notice that the troughs of the bit rate trace for $mbs = 5$ are higher in the first fifteen frames). Upon visual inspection of the sequence, this higher bit rate can most likely be attributed to the presence of an additional high detail object (a wall calendar) in the beginning of the sequence that is quickly panned out. During the $mbs = 2$ and 3 examples, there is a momentary quality loss near the beginning of the sequences, with the amount of degradation diminishing somewhat for $mbs = 3$. The level of degradation for both cases can be described simply as noticeable but not overly agitating because it occurs only once for a brief moment. In any case, it is certainly not nearly as prominent as it is for "bus."

Clearly, in the "bus" sequence, the value used for R_S is set too far below the unconstrained average bit rate. This creates a situation where a bucket overflow is inevitable. Lowering the upper threshold value below the bucket capacity will not necessarily alleviate the situation. The presence of the threshold is similar to lowering the bucket capacity with the only difference being that instead of lowering the bit rate for the present frame, accomplished through frame recoding, the bit rate for the following frame will be reduced. In situations where the SBR bucket habitually overflows, a best course of action is to recode the entire sequence using either a lower value of Q_{init}, or traffic parameters that accurately reflect the sequence's open-loop VBR statistics. There is no advantage of transmitting a sequence with a higher peak rate and lower sustained rate than the unconstrained VBR values since the end result will be a sequence with inconsistent image quality (the lower overall image quality will be interrupted by flashes of higher quality frames).

From studying table 4.3, one blaring observation is that none of the encoded sequences manages to keep the average bit rate at or below the desired value. Perhaps if the length of the sequences were sufficiently long, the average bit would eventually converge to R_S. However, the question of the length of the sequence needed to realize this is beyond the scope of this research as the amount of time and resources that would be needed would be too impractical to realize.

4.4.3 Adjusting the Maximum Burst Size

Although the purpose of a maximum burst size is to prevent successive frames from transmitting at the peak bit rate for an extended length of time, there is little fear of this occurrence with MPEG-2 video because of the I, P, B, frame ordering. Rather, the *mbs* is used to control the size of the SBR bucket and, hence, the sensitivity of the bucket when observing the average bit rate. With this in mind, some results for different values of the *mbs* are obtained. Using the values of R_P and R_S that are similar to the VBR statistics would be uninteresting because no traffic shaping would take place. Since the effects of adjusting the maximum burst size in conjunction with the sustained-bit-rate limiting has already been examined, the experiments outlined in this section will report on the effects of varying the maximum burst size subject to a peak bit rate limitation.

For arbitrary reasons, the traffic descriptors used in the middle set of each experiment in section 4.4.1 is used as the starting point for this study. From this point, two additional cases are considered, one using a higher value of *mbs* and another using a lower one. The results are displayed graphically in figure 4.9 and tabulated in table 4.5. As can be inferred from the data, the *mbs* has very limited influence when not used in conjunction with a constraining value of R_S. The only instance when the value of the maximum burst size will have an effect upon the outcome of the traffic shaped bit stream is when the value is very low. When *mbs* is equal to two, the bucket size is normally just large enough to hold an I frame. In a few cases, however, an I frame will exceed the limit of the SBR bucket and therefore, it will be coded with a slightly lower quality. This, in turn, will cause the immediately following B frames or P frames to require a higher bit rate to encoded the extra error information. Since motion-compensated frames require significantly fewer bits than intra frames, they will rarely contribute to a bucket overflow and soon the bucket occupany will recede to a lower level. Consequently, the coder can resume its normal bit rate output. This phenomenon is visible after some of the I frames for the "bus" and "calendar" sequence. The amount of rate control adjustments is only slightly greater in the $mbs = 5$ and $mbs = 10$ cases which are identical. In terms of measured PSNR, very little difference is noticeable by comparing the traces in figure 4.10. Qualitatively, there is even less of a difference between each sequence as the visual quality appears identical irrelevant of the value of *mbs*. It can be fully expected that any additional increase in *mbs* will not produce any differing results.

4.4.4 Network Induced Losses

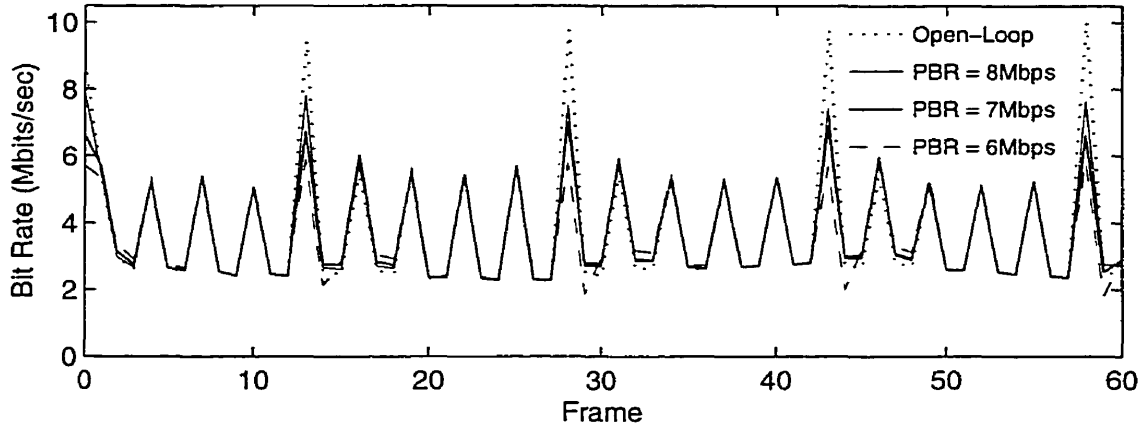
Unfortunately an actual network or a network simulator is not available to gauge the success of the traffic-shaping operation. However, since the primary goal of this study is to determine the conformance of the traffic-shaped bit stream to the network traffic descriptors using the Leaky Bucket policing algorithm, a simulation of the network response, in terms of the projected quantity of information discarded, is insightful. The complete understanding of the Leaky Bucket algorithm allows the network behavior to be predicted and a calculation of the volume of information that would be discarded by the network policer to be performed. A simulation of the network behavior is performed with the aid of the MATLAB mathematical software. Using the knowledge of the bit rates per frame of the unconstrained VBR sequences, along with the traffic descriptors, the total amount of bits that would be discarded in the absence of traffic shaping can be easily determined and is tabulated in the tables 4.7, 4.8, and 4.9. The results are listed in descending order of R_P.

The amount of forecast network loss follows the trend of the PSNR decline in the PSNR graphs. This presents the obvious conclusion that the volume of bits lost to the network policing function increases as the R_P or R_S becomes smaller and decreases as *mbs* includes more frames. As the graphs in figures 4.6 to 4.10 suggest, the majority of loss occurs during the I frames, especially during peak rate limiting, and for sustained rate limiting, losses occur in contiguous groups of frames (at the end for "bus" and near the beginning for "calendar"). This, unfortunately, implies that network losses take place in bursts and at times when they are least affordable, namely around I frames. Generally, occasional losses can be handled by the MPEG codec without any interruption of visual quality.

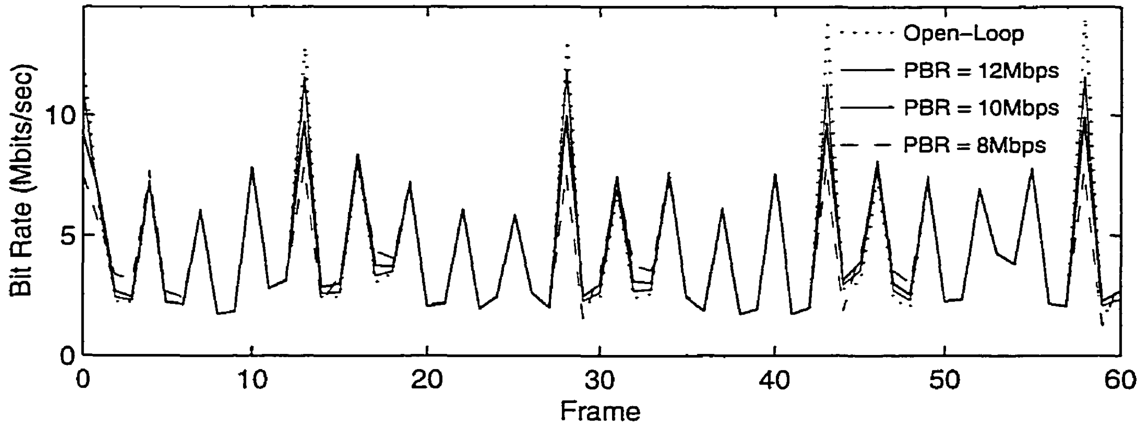
The bit loss calculated by the network policing simulation and the amount of bit reduction invoked by the rate controller for the same traffic descriptors will not be equal. The rate controller and coder can only reduce the bit rate as much as an increase in Q will allow. However, the simulation may discard any number of bits that cause a bucket overflow. In an ATM network environment, information will be packaged and handled in the context of cells rather than on an individual bit basis. This implies that network losses will be in increments of 48 bytes, the size of an ATM cell payload. Thus, more information than the values listed in the tables may be lost.

It is difficult to judge the impact of a cell loss upon the decoded sequence at the receiver. A bad situation would be the sufficient loss of information in a single frame to prevent it from being decoded and displayed. Still worse, the frame could be an

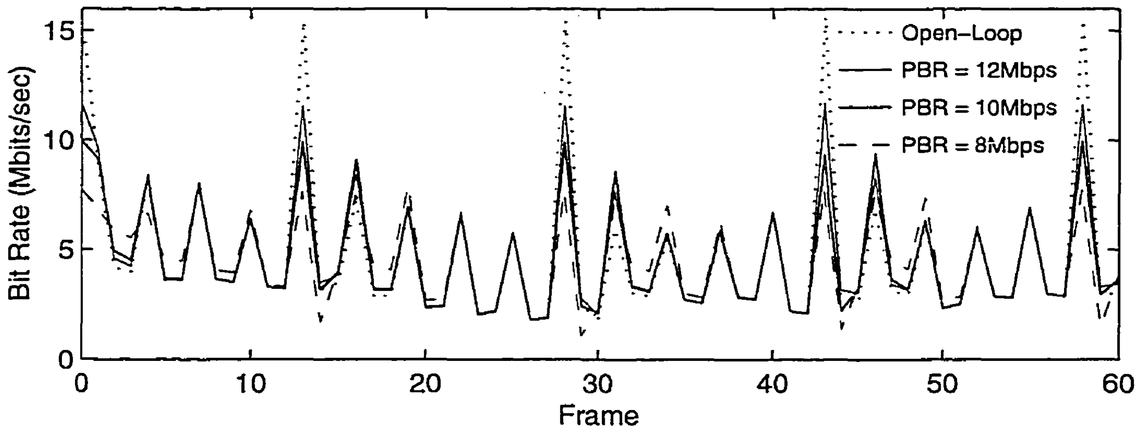
I frame in which case, the entire GOP would be corrupted. Even without any additional error correction circuitry, the MPEG-2 algorithm possesses a limited amount of error correction. For instance, MPEG-2 uses byte alignment to re-synchronize itself with the bit stream and with spatial scalability, it can mask some errors. An interesting proposal to increase the error resilience of MPEG-2 data streams is given by Kawashima et al. [32] in which they propose the elimination of I frames and replace them with "I columns" in each frame. Other methods, based on cell prioritization and dummy cell insertion, are described by Han and Orozco-Barbosa [33] and Pancha and El Zarki [34] to limit the impact of network errors. An account of the effects the Usage Parameter Control has upon MPEG-2 video in an simulated ATM network environment is provided by Richardson and Riley [27]. Specifically, they examine the effects of cell losses on MPEG coded video data.



(a) "bus"



(b) "flower garden"



(c) "calendar"

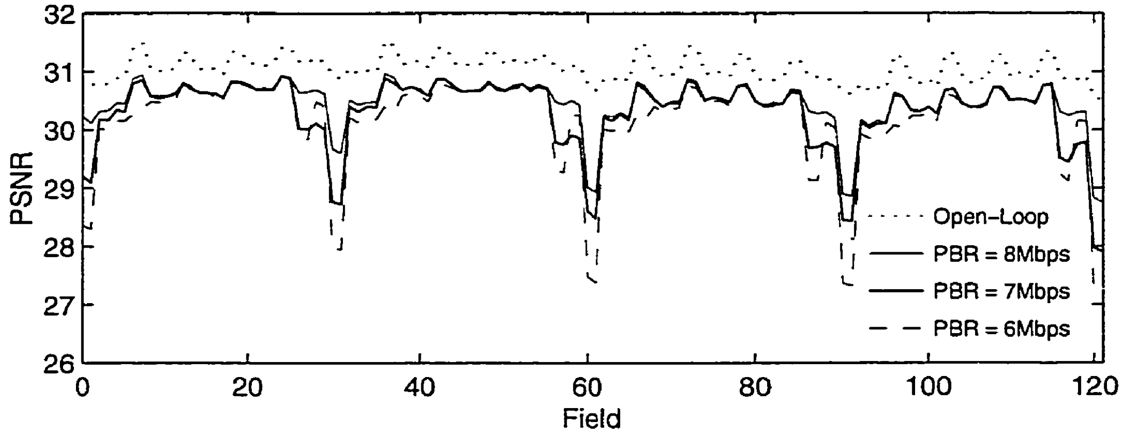
Fig. 4.5 Bit rate profiles for PBR traffic shaping

Traffic Descriptors			*Measured Statistics*				
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (Mbps)*	*avg (Mbps)*	*min (Mbps)*	*burstiness*	*std (Mbps)*
Sequence: "bus"							
8.0000	3.8482	5	7.9531	3.7719	2.2670	2.1085	1.7088
7.0000	3.8482	5	6.9955	3.6867	2.2039	1.8975	1.5865
6.0000	3.8482	5	5.9100	3.6411	1.8264	1.6231	1.4217
Sequence: "flower garden"							
12.0000	4.4462	5	11.8661	4.3924	1.7179	2.7015	2.9782
10.0000	4.4462	5	9.9607	4.3548	1.7290	2.2873	2.6424
8.0000	4.4462	5	7.9860	4.2337	1.2314	1.8863	2.3347
Sequence: "calendar"							
12.0000	4.8884	5	11.7274	4.7542	1.7878	2.4668	2.8479
10.0000	4.8884	5	9.9701	4.6458	1.8014	2.1460	2.5831
8.0000	4.8884	5	7.9006	4.5298	1.0862	1.7441	2.1352

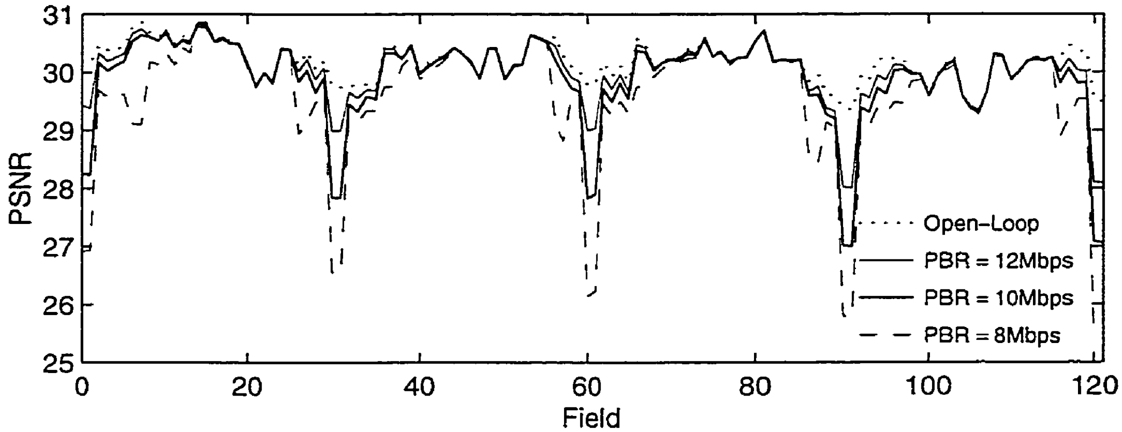
Table 4.1 Bit rate statistics for PBR traffic shaping

Traffic Descriptors			*Measured Statistics*			
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (dB)*	*avg (dB)*	*min (dB)*	*std (dB)*
Sequence: "bus"						
8.0000	3.8482	5	30.9603	30.4665	28.7466	0.4329
7.0000	3.8482	5	30.9282	30.3047	27.8953	0.6263
6.0000	3.8482	5	30.9002	30.1457	27.2454	0.8383
Sequence: "flower garden"						
12.0000	4.4462	5	30.8558	30.0398	28.0054	0.5399
10.0000	4.4462	5	30.8362	29.8879	27.0041	0.7855
8.0000	4.4462	5	30.7875	29.5638	25.4937	1.1345
Sequence: "calendar"						
12.0000	4.8884	5	29.4454	28.6492	25.8887	0.8823
10.0000	4.8884	5	29.4427	28.5102	24.6243	1.1339
8.0000	4.8884	5	29.3823	28.0588	23.6944	1.4509

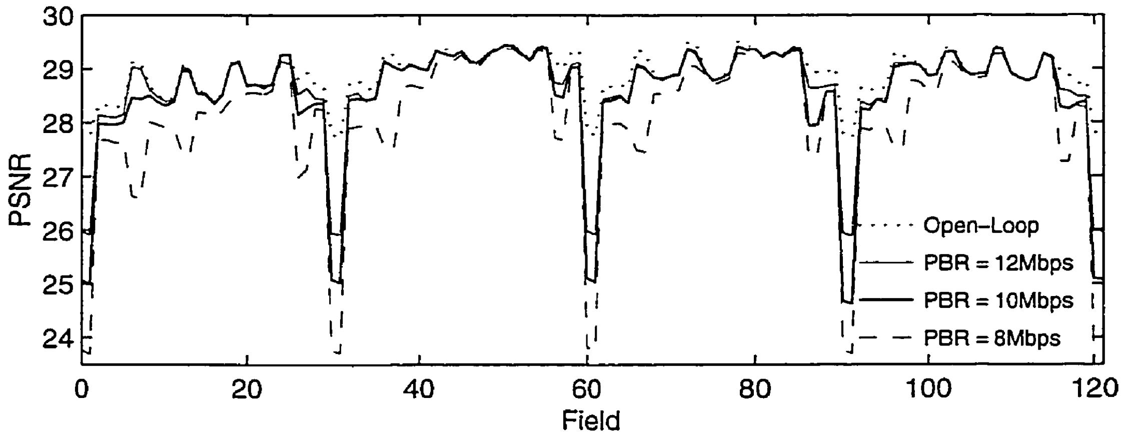
Table 4.2 Luminance PSNR statistics for PBR traffic shaping



(a) "bus"

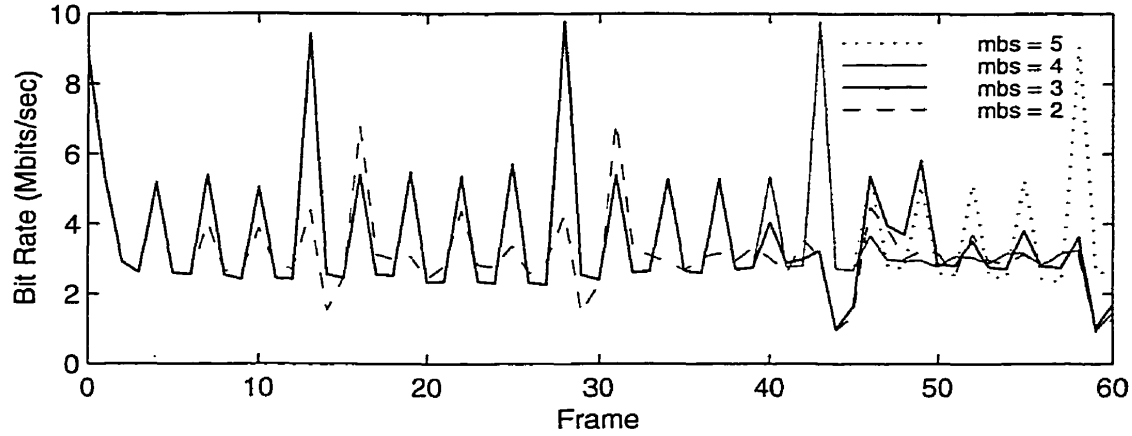


(b) "flower garden"

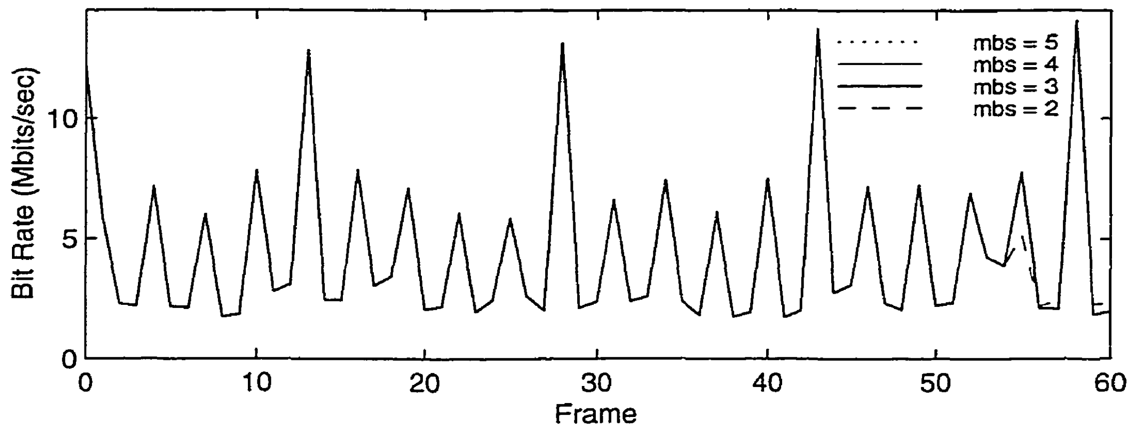


(c) "calendar"

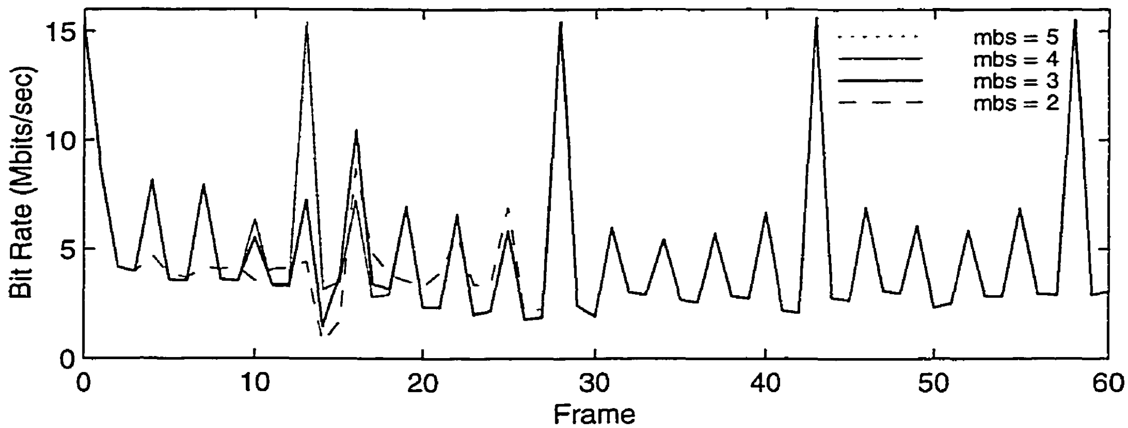
Fig. 4.6 Luminance PSNR curves for PBR traffic shaping



(a) "bus"



(b) "flower garden"

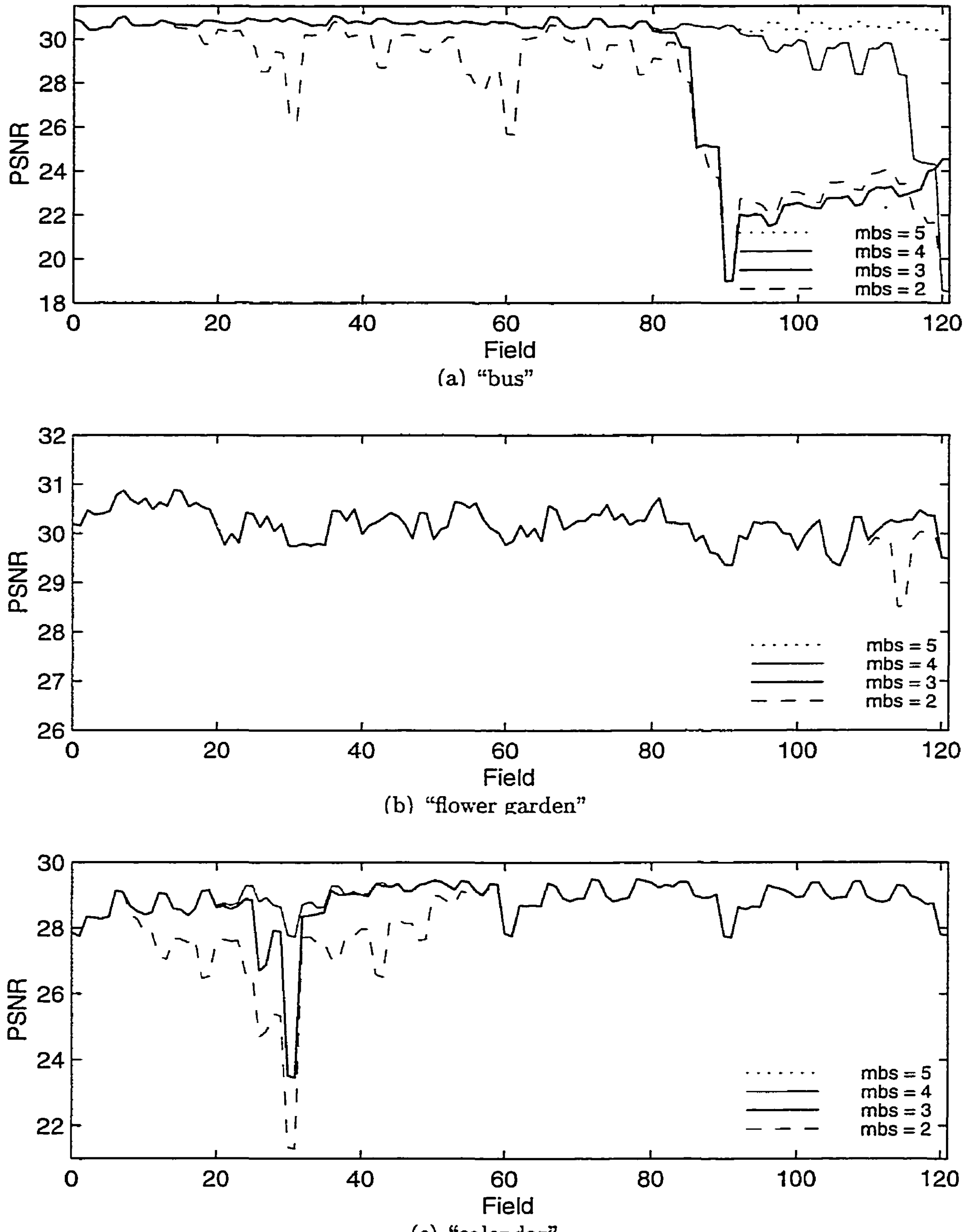


(c) "calendar"

Fig. 4.7 Bit rate profiles for SBR traffic shaping

Traffic Descriptors			*Measured Statistics*				
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (Mbps)*	*avg (Mbps)*	*min (Mbps)*	*burstiness*	*std (Mbps)*
Sequence: "bus"							
10.0286	3.0000	2	8.8301	3.2005	0.8938	2.7590	1.3178
10.0286	3.0000	3	9.7858	3.5613	0.9895	2.7478	1.8211
10.0286	3.0000	4	9.7992	3.6185	0.9245	2.7081	1.9372
10.0286	3.0000	5	9.7992	3.8357	2.2606	2.5547	2.0704
Sequence: "flower garden"							
14.0688	4.0000	2	14.0688	4.4237	1.7206	3.1803	3.3141
14.0688	4.0000	3	14.0688	4.4462	1.7206	3.1642	3.3546
14.0688	4.0000	4	14.0688	4.4462	1.7206	3.1642	3.3546
14.0688	4.0000	5	14.0688	4.4462	1.7206	3.1642	3.3546
Sequence: "calendar"							
15.6427	4.0000	2	15.6427	4.6431	0.7073	3.3690	3.3388
15.6427	4.0000	3	15.6427	4.7912	1.4674	3.2649	3.4980
15.6427	4.0000	4	15.6427	4.8884	1.8072	3.2000	3.6765
15.6427	4.0000	5	15.6427	4.8884	1.8072	3.2000	3.6765

Table 4.3 Bit rate statistics for SBR traffic shaping



(a) "bus"

(b) "flower garden"

(c) "calendar"

Fig. 4.8 Luminance PSNR curves for SBR traffic shaping

Traffic Descriptors			*Measured Statistics*			
R_P *(Mbps)*	R_S *(Mbps)*	*mbs* *(frames)*	*max* *(dB)*	*avg* *(dB)*	*min* *(dB)*	*std* *(dB)*
Sequence: "bus"						
10.0286	3.0000	2	31.0285	27.5925	18.8079	3.4345
10.0286	3.0000	3	31.0693	28.3685	18.9887	3.6764
10.0286	3.0000	4	31.0693	30.0541	18.5041	1.9351
10.0286	3.0000	5	31.0693	30.6547	29.7893	0.2027
Sequence: "flower garden"						
14.0688	4.0000	2	30.8816	30.1392	28.5038	0.4036
14.0688	4.0000	3	30.8816	30.1890	29.3449	0.3403
14.0688	4.0000	4	30.8816	30.1890	29.3449	0.3403
14.0688	4.0000	5	30.8816	30.1890	29.3449	0.3403
Sequence: "calendar"						
15.6427	4.0000	2	29.5063	28.2302	21.2960	1.3315
15.6427	4.0000	3	29.5063	28.7551	23.4593	0.8586
15.6427	4.0000	4	29.5063	28.8972	27.7159	0.4538
15.6427	4.0000	5	29.5063	28.8972	27.7159	0.4538

Table 4.4 Luminance PSNR statistics for SBR traffic shaping

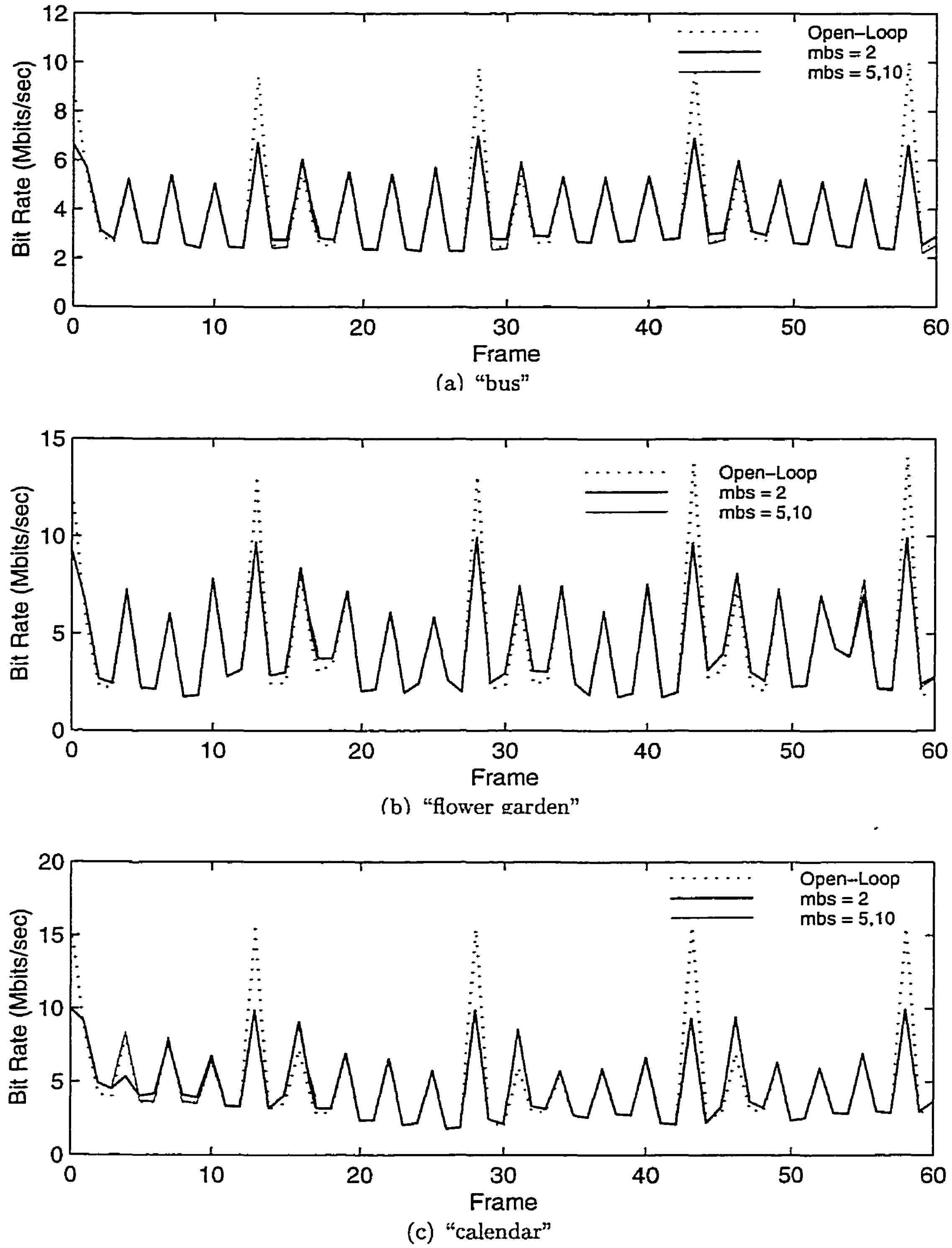


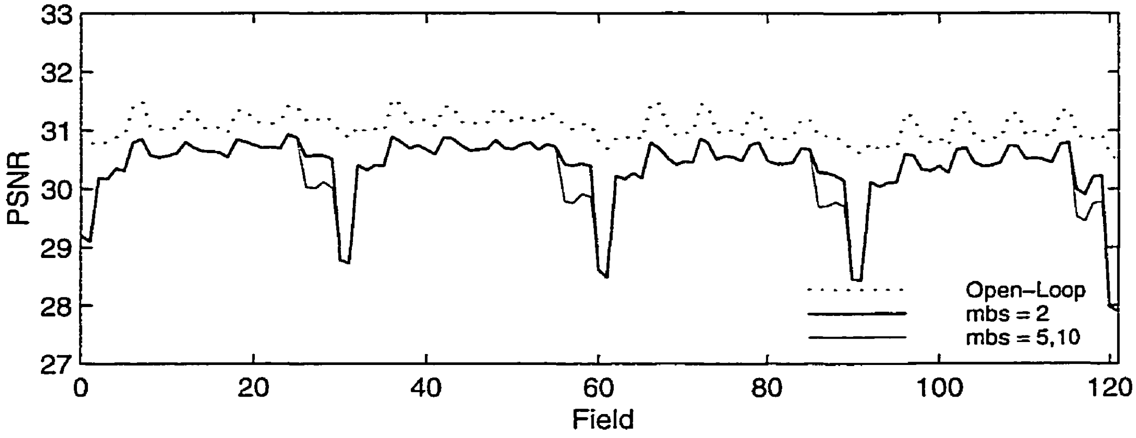
Fig. 4.9 Bit rate profiles for difference values of mbs

Traffic Descriptors			Measured Statistics				
R_P (Mbps)	R_S (Mbps)	mbs (frames)	max (Mbps)	avg (Mbps)	min (Mbps)	burstiness	std (Mbps)
Sequence: "bus"							
7.0000	3.8482	2	6.9955	3.7334	2.2867	1.8737	1.5541
7.0000	3.8482	5	6.9955	3.6867	2.2039	1.8975	1.5865
7.0000	3.8482	10	6.9955	3.6867	2.2039	1.8975	1.5865
Sequence: "flower garden"							
10.0000	4.4462	2	9.9607	4.3477	1.7290	2.2910	2.6226
10.0000	4.4462	5	9.9607	4.3548	1.7290	2.2873	2.6424
10.0000	4.4462	10	9.9607	4.3548	1.7290	2.2873	2.6424
Sequence: "calendar"							
10.0000	4.8884	2	9.9701	4.6281	1.8014	2.1543	2.5247
10.0000	4.8884	5	9.9701	4.6458	1.8014	2.1460	2.5831
10.0000	4.8884	10	9.9701	4.6458	1.8014	2.1460	2.5831

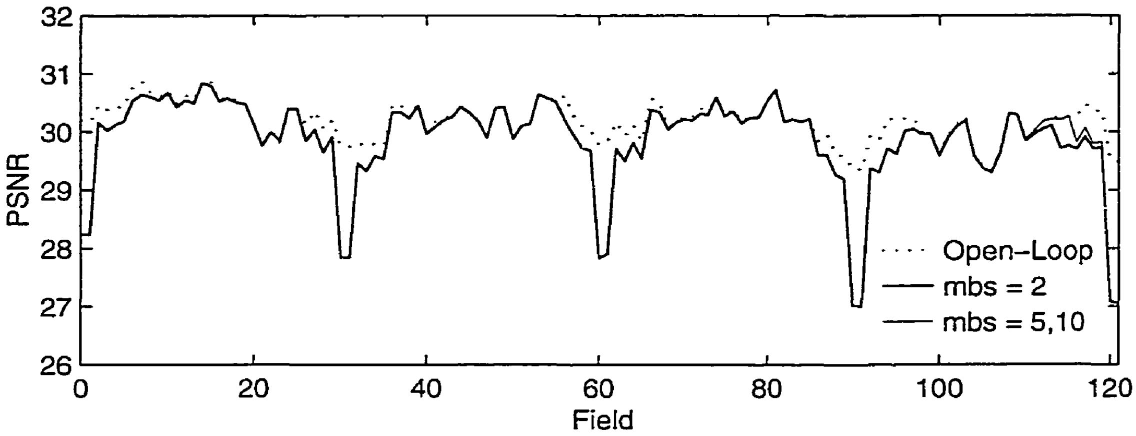
Table 4.5 Bit rate statistics for different values of mbs

Traffic Descriptors			Measured Statistics			
R_P (Mbps)	R_S (Mbps)	mbs (frames)	max (dB)	avg (dB)	min (dB)	std (dB)
Sequence: "bus"						
7.0000	3.8482	2	30.9282	30.3722	27.8953	0.5958
7.0000	3.8482	5	30.9282	30.3047	27.8953	0.6263
7.0000	3.8482	10	30.9282	30.3047	27.8953	0.6263
Sequence: "flower garden"						
10.0000	4.4462	2	30.8362	29.8734	27.0041	0.7838
10.0000	4.4462	5	30.8362	29.8879	27.0041	0.7855
10.0000	4.4462	10	30.8362	29.8879	27.0041	0.7855
Sequence: "calendar"						
10.0000	4.8884	2	29.4427	28.4487	24.6243	1.1411
10.0000	4.8884	5	29.4427	28.5102	24.6243	1.1339
10.0000	4.8884	10	29.4427	28.5102	24.6243	1.1339

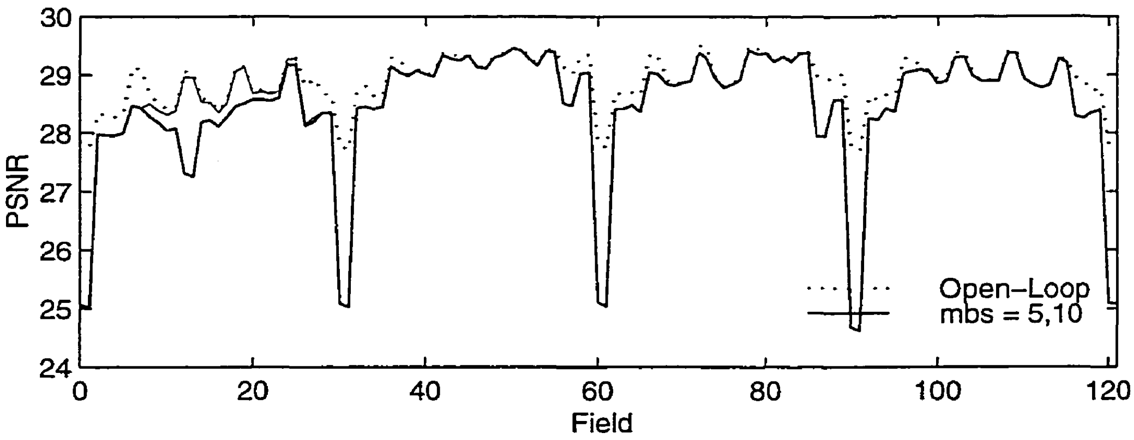
Table 4.6 Luminance PSNR statistics for different values of mbs



(a) "bus"



(b) "flower garden"



(c) "calendar"

Fig. 4.10 Luminance PSNR curves for different values of mbs

Traffic Descriptors			*Projected Losses*	
$R_P(Mbps)$	$R_S(Mbps)$	*mbs(frames)*	*bits*	*percentage*
10.0286	3.0000	2	771128	9.8552
10.0286	3.0000	3	488400	6.2418
10.0286	3.0000	4	240177	3.0695
10.0286	3.0000	5	8732	0.1116
8.0000	3.8482	5	262998	3.3612
7.0000	3.8482	2	429663	5.4912
7.0000	3.8482	5	429663	5.4912
7.0000	3.8482	10	429663	5.4912
6.0000	3.8482	5	596333	7.6212

Table 4.7 Projected network bit losses for "bus"

Traffic Descriptors			*Projected Losses*	
$R_P(Mbps)$	$R_S(Mbps)$	*mbs(frames)*	*bits*	*percentage*
14.0688	4.0000	2	74931	0.8288
14.0688	4.0000	3	0	0
14.0688	4.0000	4	0	0
14.0688	4.0000	5	0	0
12.0000	4.4462	5	200669	2.2197
10.0000	4.4462	2	545489	6.0338
10.0000	4.4462	5	533999	5.9067
10.0000	4.4462	10	533999	5.9067
8.0000	4.4462	5	867334	9.5938

Table 4.8 Projected network bit losses for "flower garden"

Traffic Descriptors			*Projected Losses*	
$R_P(Mbps)$	$R_S(Mbps)$	*mbs(frames)*	*bits*	*percentage*
15.6427	4.0000	2	537090	5.4035
15.6427	4.0000	3	296180	2.9798
15.6427	4.0000	4	1	10^{-5}
15.6427	4.0000	5	1	10^{-5}
12.0000	4.8884	5	587853	5.9142
10.0000	4.8884	2	921183	9.2677
10.0000	4.8884	5	921183	9.2677
10.0000	44.8884	10	921183	9.2677
8.0000	4.8884	5	1279258	12.8702

Table 4.9 Projected network bit losses for "calendar"

Chapter 5

Conclusion

At the time of completion of this thesis, a major entry into the consumer electronics market, the Digital Video Disc (DVD), had recently been introduced in North America. The major selling point of the DVD is its ability to provide amazingly clear full-screen video with the aid of the MPEG-2 compression algorithm. This product has attracted much attention and is a sign of the growing movement pushing for the digitization of the video medium. The progression of this technology will eventually see video data sent to individual clients or broadcast over broadband networks as common occurrences. However, with transmission of video in digital formats comes the associated problem of reliable and efficient transmission. To ensure the proper reception of the source data at the receiver, traffic shaping must be involved. In particular, a thorough understanding of the operation of the network policing mechanism, especially for an ATM network, would be beneficial. This is the central topic investigated in this thesis.

5.1 Thesis Summary

Prior to the examination of the effects of traffic shaping, results contrasting the differences between constant bit rate (CBR) and variable bit rate (VBR) have been presented. To generate CBR bit streams, a traffic-shaping coder is used. Encoding at a constant bit rate using the traffic-shaping encoder is not as simple as equating the target peak bit rate with the sustained bit rate. Since these target rates only serve as upper bounds in the definition of the traffic-shaping function, the actual average bit rates tend to be lower than desired. One method to compensate for this shortcoming is to set the target bit rate to a higher value than the desired rate. Unfortunately, the determination of the amount of this increase is essentially a trial and error exercise. Achieving constant bit rate in this manner has an advantage over more established methods, such as the MPEG-2 *Test Model 5*, because the bit rate can be held constant over a much finer interval. More precisely, while the TM5 is only capable of maintaining bit rate consistency when considering an entire group of pictures, the method explored in this thesis can produce constant bit rates between successive frames. Practically, however, holding the rate constant over just am single frame should never be done. A much more sensible approach is to average the bit rate over multiple frames so that the bit rate variations can be shared by several frames. This has the benefit that the stability of the image quality, both perceptually and numerically (using the peak signal to noise ratio as the evaluation criterion), increases with the length of the averaging interval, n. However, if the average interval is a multiple of a GOP, then the method outlined by the MPEG-2 Test Model 5 provides vastly superior results. The TM5 encoder is able to generate results whose quality nearly matches those observed from pure VBR coded sequences, all while maintaining a constant bit rate over a group of pictures. The limitation for TM5 encoders, obviously, is that the averaging interval is always a multiple of a M (the length of a GOP). Discounting the initial GOP, the TM5 is able to remain very close to the desired bit rate. This is unlike the rate-constrained coder which tends to fluctuate above and below the average. Whereas the sequences generated from the Test Model 5 coder possesses image quality comparable to that of pure VBR sequences, the rate-constrained CBR sequences may exhibit considerable quality loss. However, the degree of loss diminishes as n increases. Another disadvantage that the rate-constrained coder suffers when trying to generate constant-rate-bit streams is the lack of precision in specifying the exact output bit rate. Instead, only the value of the upper limit may be defined.

The Leaky Bucket algorithm is incorporated into the coder's rate controller in order to maximize the likelihood that the rate controller will be able to detect the exact instances at which the ATM network policer (which also uses the Leaky Bucket) would detect a traffic violation. The algorithm relies upon three traffic descriptors to determine whether the current frame bits will meet the approval of the network. These descriptors are called the peak bit rate (PBR), sustained bit rate (SBR), and maximum burst size (mbs). The original Leaky Bucket (also known as the Generic Cell Rate Algorithm, GCRA) is defined for use in an ATM environment and therefore uses cells instead of bits in its definition. Through a slight modification of the definition of the GCRA, the algorithm is fit into the context of an off-line, bit-rate-based traffic monitoring function.

When limiting the peak rate, specifying rates that are up to 20% below the open-loop peak bit rate is found to provide image quality rivaling that of unconstrained VBR equivalents. Experimental evidence suggests that for the sequences considered in this study, each 1Mbps decrease in R_P (the declared peak bit rate) roughly translates into a 0.2dB loss in average PSNR which is not a visually discernible change in quality. However, while the overall average may not change significantly, the variation in the perceived quality starts to become increasingly noticeable. Reductions of up to nearly 50% of the unconstrained peak bit rate have been examined and the viewed images clearly show the strain of restricting the bit rate by the presence of a high degree of quality inconsistency. The image PSNR tends to drop around the display of each I frame and this creates video that may be perceptually irritating. Rather than placing such a large a restriction upon the bit rate, a better option would be to recode the entire sequence with a lower bit rate using a higher initial value of the quantization scaling factor, Q; this would lower the overall quality of the decoded sequence, but the image consistency would improve.

While the Leaky Bucket seems to be adept at policing the peak bit rate, handling the sustained bit rate appears to be a difficult task. The source of this problem is the long reaction time the SBR Leaky Bucket. Generally, the bucket fills up very slowly and until the capacity is exceeded, no corrective measure is taken. However, the sensitivity to the sustained bit rate may be increased by decreasing the bucket size through modifications to the value of *mbs*. The experimental results indicate that the effects of sustained bit rate limitations range from severe image degradation (for the "bus" sequence with the sustained rate, R_S, 22.0% below the measured open- loop average rate) to virtually no difference when compared to the open-loop results (for the "flower garden" sequence with the value of R_S 10.0% below the open-loop average rate). Special attention must be taken when attempting to restrict the average bit rate significantly below the unconstrained value to prevent excessive picture quality loss. Of course, the time of occurrence and the severity of the degradation depend upon the image complexity and activity as well as the value of R_S. Assuming that the amount of detail and motion remain fairly constant over the duration of the video sequence, increasing the bucket capacity may not help prevent overflows but rather, serve to delay the corrective efforts of the rate controller. It is difficult to make an evaluation as to the degree of acceptable values for the sustained bit rate, R_S, because in all the conducted experiments, the measured average bit rate always exceeded the target value. Longer sequences are needed to determine whether the sustained bit rate is attained at a much later time or else the algorithm is deficient in this respect. However, based upon the findings of this thesis, a recommendation can be made that R_S should be defined to be a value close to the measured VBR average bit rate. Even the use of values just 22% lower have resulted in lengthy periods of image degradation. This is considerably worse than was the case for PBR limiting in which only a few frames were typically affected with moderate to no noticeable image quality change.

The maximum burst size (mbs) traffic descriptor is more important for its influence upon the SBR bucket rather than as a parameter to control the duration of bursts at the peak bit rate. For MPEG-2 sequences, the ordering of the I, P, and B frames makes peak bit rate bursts greater than the duration of a single frame nonexistent. While a smaller value of mbs may cause the involvement of the traffic shaping function to increase slightly, very little impact is felt in the decoded image qualities. Subjective impressions comparing the image qualities of sequences encoded with $mbs = 2, 5$ and 10 reveal that each set of sequences are indiscernible from each other.

The unfortunate situation associated with transmitting MPEG-2 data over a network that uses the Leaky Bucket algorithm in its policing system is that network packet losses tends to occur in bursts around I frames. This reinforces the imperative need to traffic shape the data before transmitting across the network.

5.2 Future Work

A major limitation in the advancement of work in this thesis has been the software MPEG-2 codec. This was expected since the initial intent for developing a software MPEG-2 codec at INRS was for educational and heuristic purposes. As such, the coder, while perfectly capable of generating valid MPEG-2 files, lacks efficiency in coding and suffers from long processing times. However, given the rapid increase in CPU processing speed, this is bound to become less of an issue.

This thesis has touched upon many issues pertinent to the transmission of video over ATM networks. Many issues extending from the work presented here are left to be explored. A natural extension of this research would be to investigate the actual performance of the traffic shaping in a real, or simulated, network environment and then to study the effects that the network information loss has upon the received video signal. This would be extremely helpful in validating the assumption that the Leaky Bucket employed in the coder's rate controller is truly successful in predicting the behavior of the network's policing function. To create a even more realistic situation, a video file server should be included in the simulations.

Another insightful continuation of this project would be to conduct experiments that measure the effects of rate constraining one or more of the MPEG-2 scalable layers upon the resultant image quality. Ghanbari and Azari [35] have already investigated the performance of sending two-layer video signals over an ATM network in terms of network delay and loss. However, traffic shaping was not considered in their research. In addition, their work did not extend to exploring the qualitative or quantitative picture quality of the received video signal.

With respect to the sustained-bit-rate traffic shaping, many avenues of research are still possible. One would be to devise a method to determine or estimate the amount of time that must pass before the rate controller achieves its objective and forces the average bit rate below the target value. Another interesting study could be to examine the effects the placing upper and lower thresholds in the bucket. The location of the thresholds and the width of the hysteresis they provide could be studied to examine their effects upon the coding efficiency and quality of the resulting compressed video sequence.

This thesis is a progression in the rapidly evolving field of packet video. The amount of work that remains to be performed to lead this technology to maturity is still considerable. While digital video, especially with MPEG-2, is already firmly established, ATM is still in its infancy. MPEG-2 has already permeated consumer electronics but an extensive ATM network has yet to be realized. However, when this event does happen, many more areas of discussion for packet video will be forthcoming.

Appendix A

Definition of Terms and Variables

(Variable names are shown in italics.)

Asynchronous Transfer Mode a network transport protocol using fixed-sized packets, called cells, to transmit data over a fixed path through the network.

ATM see *Asynchronous Transfer Mode*

bi-directional (B) frame an MPEG-2 frame that uses motion compensation from the previous and next reference frame.

burst length see *maximum burst size.*

burstiness a metric used to evaluate the variability of data transmission rate, equivalent to the ratio of the peak rate to the average rate.

CBR constant bit rate.

frate a variable that defines the display frame rate (in frames per second).

GCRA see *Generic Cell Rate Algorithm.*

Generic Cell Rate Algorithm the name given to the Leaky Bucket traffic policing algorithm by the ATM Forum.

GOP see *group of pictures*

group of pictures a sequence of M consecutive frames extending from an I frame to the next I frame.

intra (I) frame an MPEG-2 frame that uses spatial compression only.

K_B the ratio of the Q used for B frames to the Q used for I frames.

K_P the ratio of the Q used for P frames to the Q used for I frames.

Leaky Bucket an algorithm used to policing network traffic through a method conceptually akin to data packets filling a fixed sized bucket while draining out at a steady pace.

M a variable in MPEG-2 that defines the distance (in frames) between consecutive P frames.

maximum burst size a traffic shaping parameter that specifies the longest period during which the peak transmission rate can be sustained.

mbs a variable that defines the *maximum burst size* (in number of frames).

motion compensation a method of interpolating the image content of a frame through the use of motion vectors from a reference frame.

MPEG-2 the second phase of the video compression algorithm developed by the ISO/IEC Moving Pictures Experts Group.

N a variable in MPEG-2 that defines the distance (in frames) between consecutive I frames.

n a variable that defines the number of frames over which the bit rate will be averaged.

open-loop coder a source coder without any rate constraints whose primary goal is to maintain the consistency of the decoded image quality. (same as unconstrained coder)

packet video the transfer of video signals over asynchronously time division multiplexed networks such as ATM.

PBR peak bit rate.

policing a network function that detects non-conforming sources and takes appropriate actions to minimize the potentially negative effect to the conforming sources.

predictive (P) frame an MPEG-2 frame that uses motion compensation from the previous reference frame.

Q the quantization scaling factor.

Q_{init} the initial value assigned to the quantization scaling factor.

quality of service a set of transmission quality assurances made by an ATM network such as cell loss rate, maximum delay, and maximum delay jitter.

QoS see *quality of service.*

reference frame the most recent or upcoming I or P frame that will serve as the reference for motion compensation.

SBR sustained (average) bit rate.

R_P a variable that defines the peak bit rate (in bits per second).

R_S a variable that defines the sustained bit rate (in bits per second).

Test Model 5 a coding model of the MPEG-2 algorithm used during the evaluation and verification stages of the standard.

TM5 see *Test Model 5.*

traffic shape the process by which the time varying transmission rate of a source is adjusted to meet the expectations of the network's policing function.

Usage Parameter Control the joint operation of a rate controller based upon traffic descriptors and the network policing function.

VBR variable bit rate.

Appendix B

Chrominance Graphs

This appendix contains the graphical representations of the PSNR values for the chrominance components of the images considered in the experiments. The organization is as follows:

1. **Variable Bit Rate** (Chapter 3)

   Figures B.1 (page 100) to B.2 (page 100).

2. **Constant Bit Rate** (Chapter 3)

   Two sets of constant bit rate experimental results are provided — one for the results obtained through the use of the rate constrained coder and the other through the use of the Test Model 5 coder.

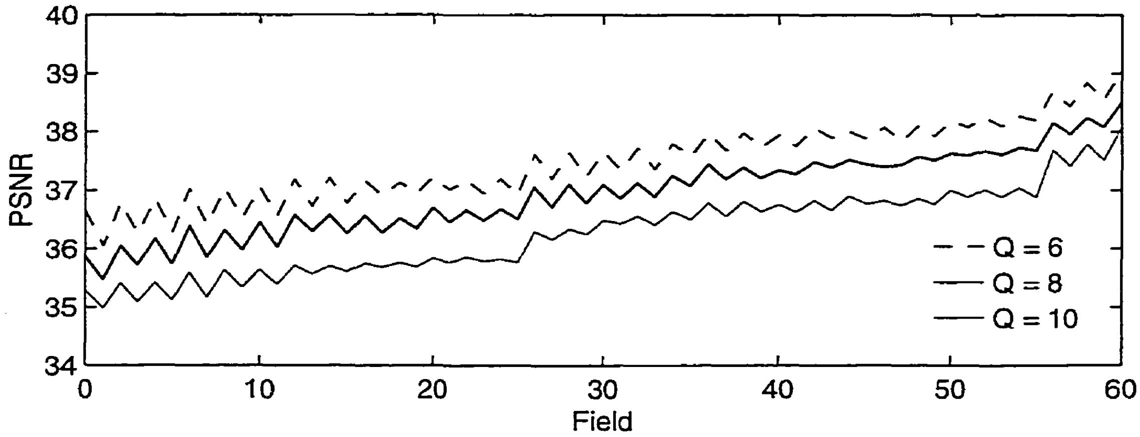
   - *Rate-Constrained Coder:* Figures B.3 (page 102) to B.4 (page 102).
   - *Test Model 5:* Figures B.5 (page 104) to B.6 (page 104).

3. **Traffic Shaping** (Chapter 4)

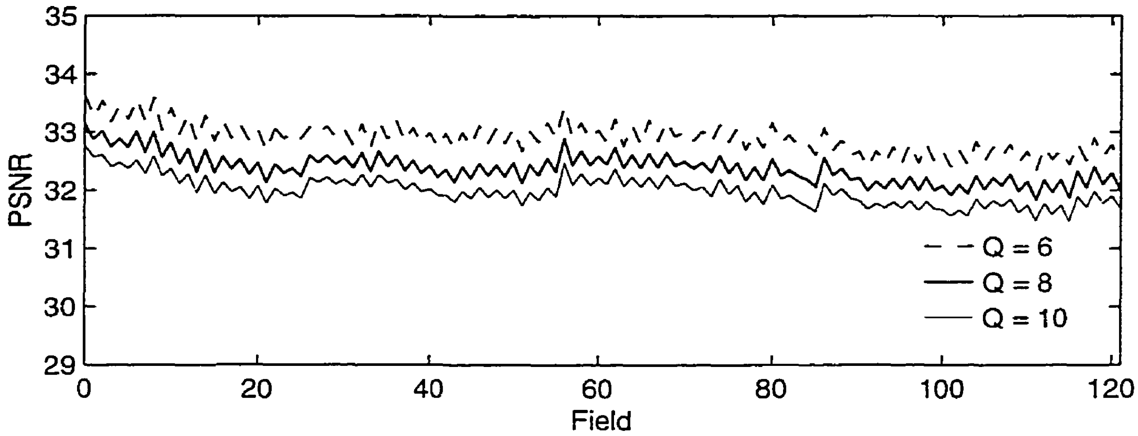
   Three different sets of experimental results involving traffic shaping are provided — peak bit rate limiting, sustained bit rate limiting, and maximum burst size adjusting.

   - *Peak-Bit-Rate Limiting* (Section 4.4.1): Figures B.7 (page 106) to B.8 (page 106).

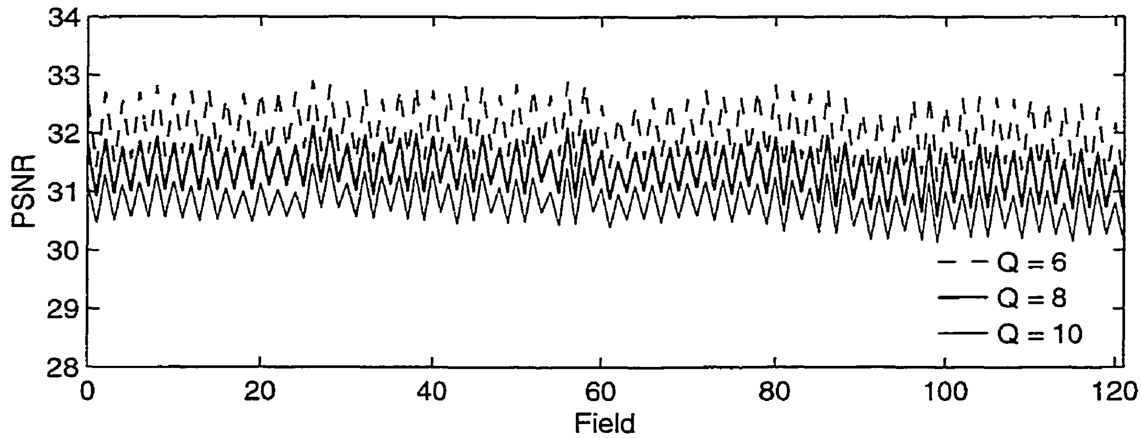
- *Sustained-Bit-Rate Limiting* (Section 4.4.2): Figures B.9 (page 108) to B.10 (page 108).
- *Maximum-Burst-Size Adjusting* (Section 4.4.3): Figures B.11 (page 110) to B.12 (page 110).



(a) "bus"

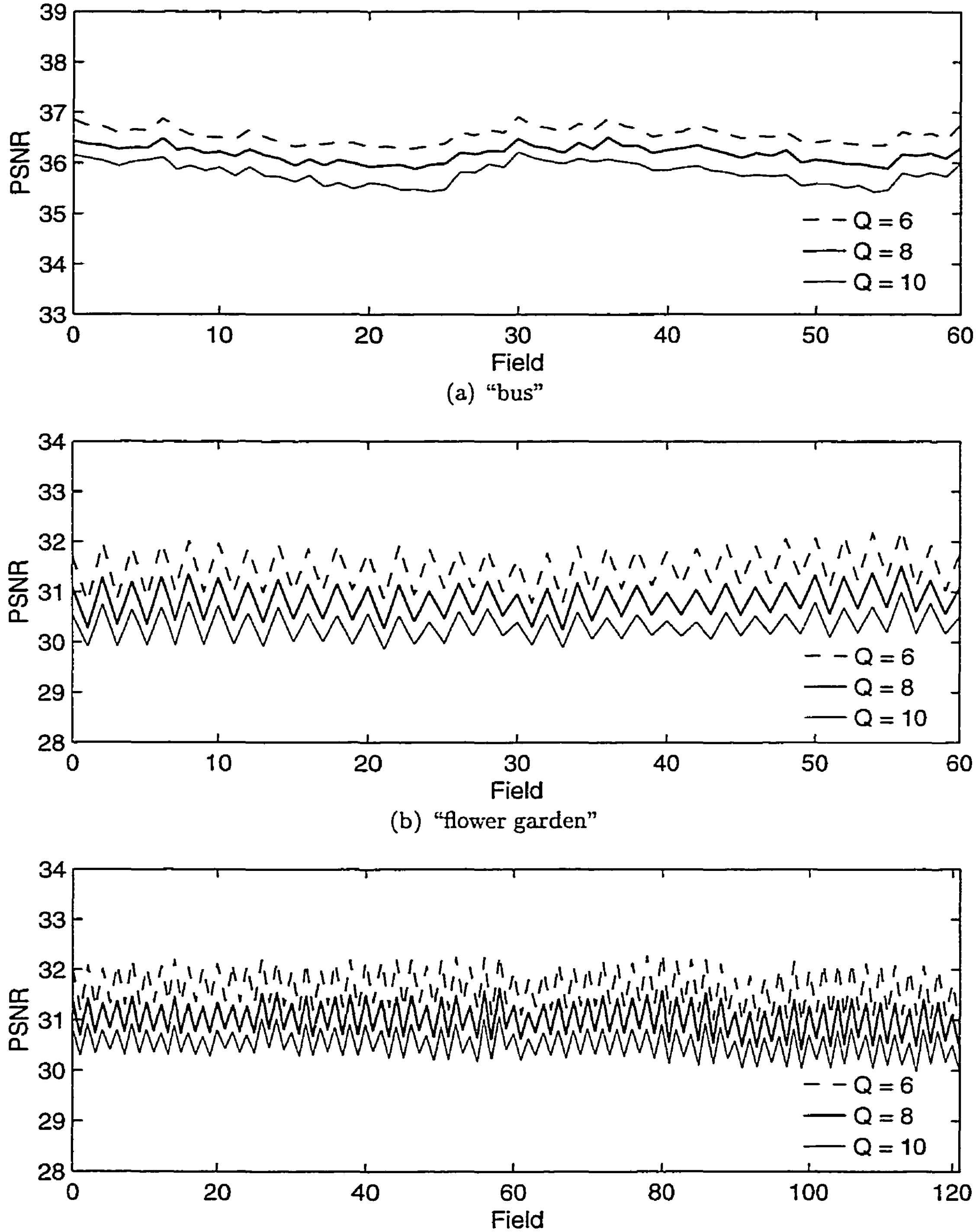


(b) "flower garden"



(c) "calendar"

Fig. B.1 VBR Chrominance-1 PSNR curves for different values of Q

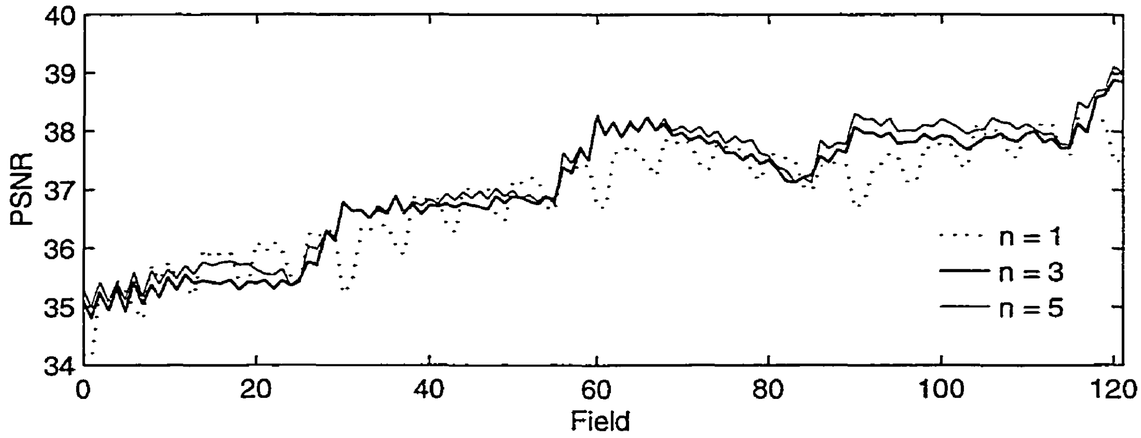


(a) "bus"

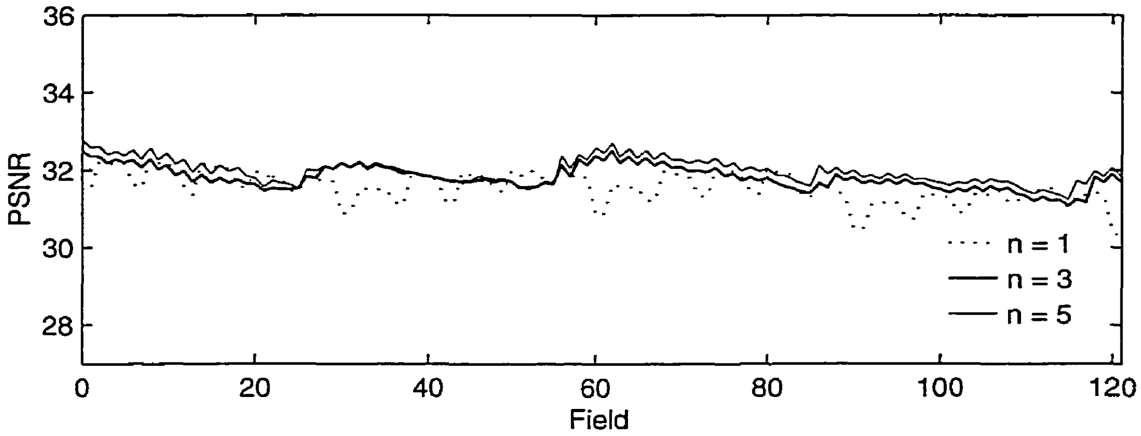
(b) "flower garden"

(c) "calendar"

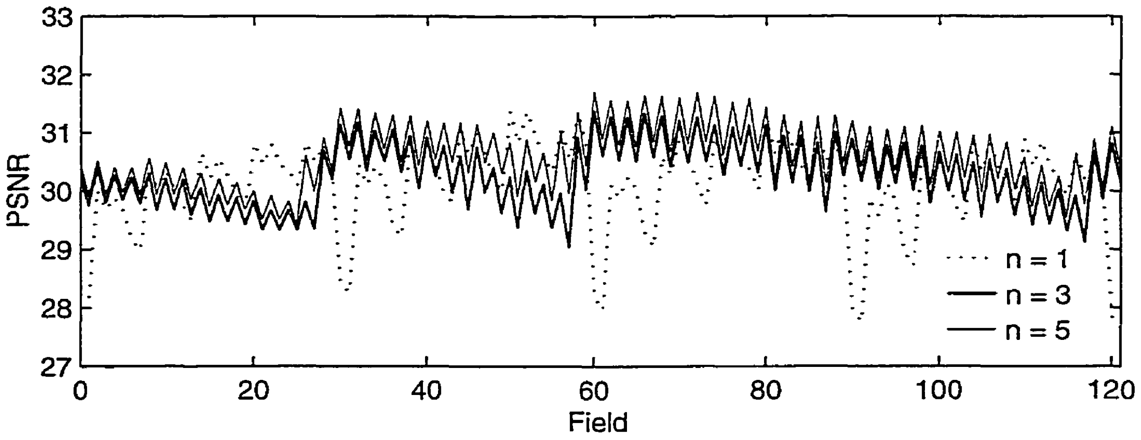
Fig. B.2 VBR Chrominance-2 PSNR curves for different values of Q



(a) "bus"

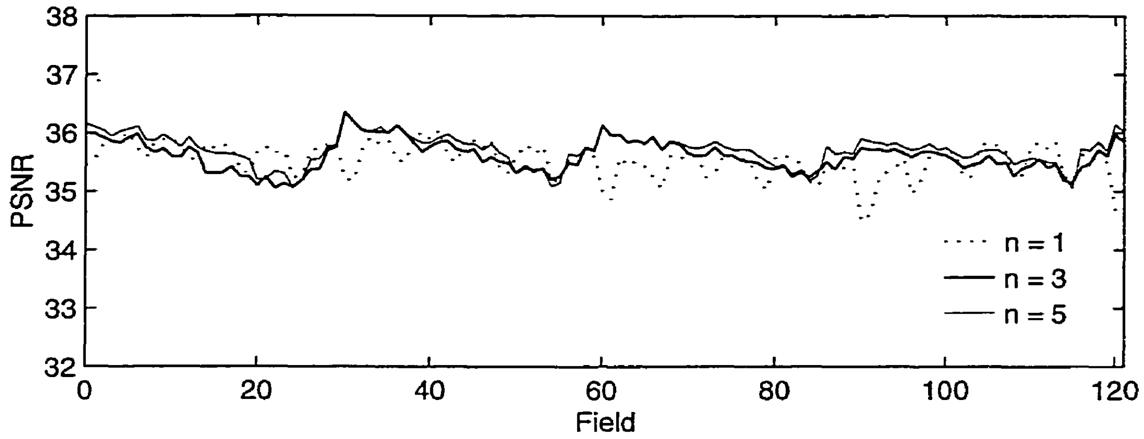


(b) "flower garden"

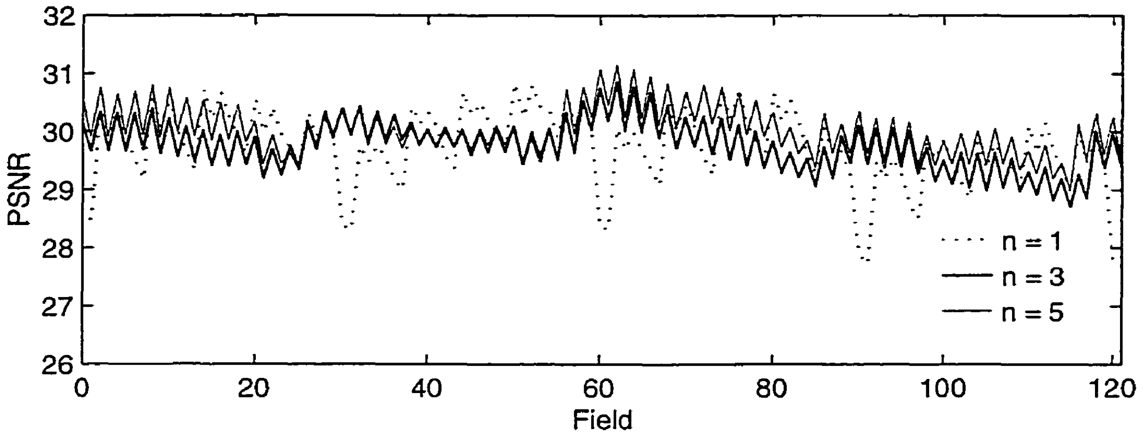


(c) "calendar"

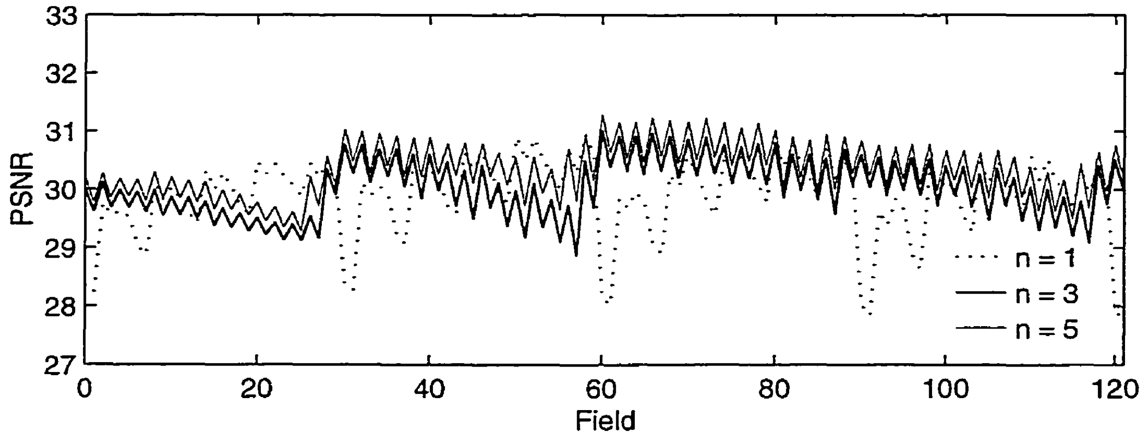
Fig. B.3 CBR chrominance-1 PSNR curves for different values of Q



(a) "bus"



(b) "flower garden"



(c) "calendar"

Fig. B.4 CBR chrominance-2 PSNR curves for different values of Q

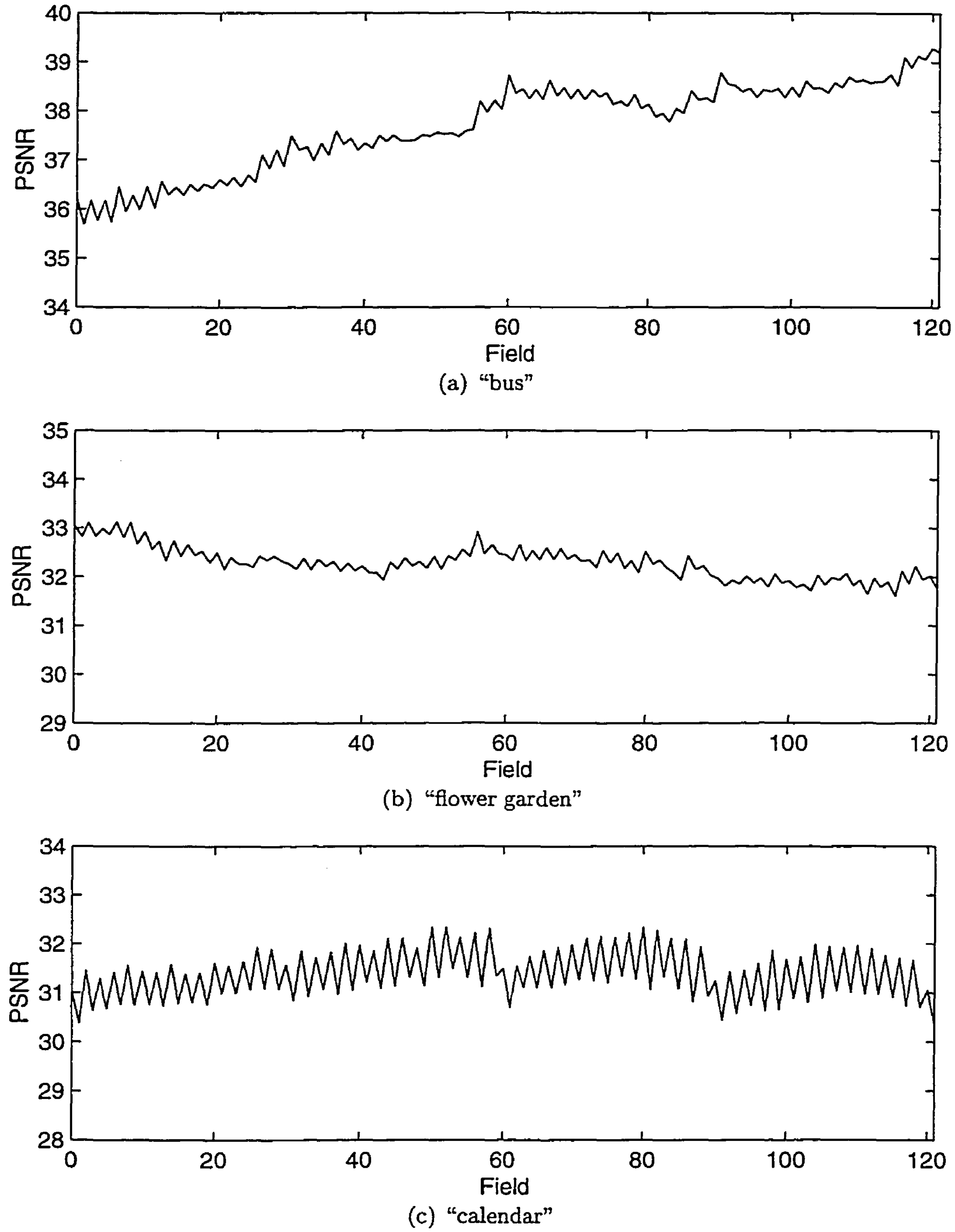


Fig. B.5 TM5 chrominance-1 PSNR curves for different values of Q

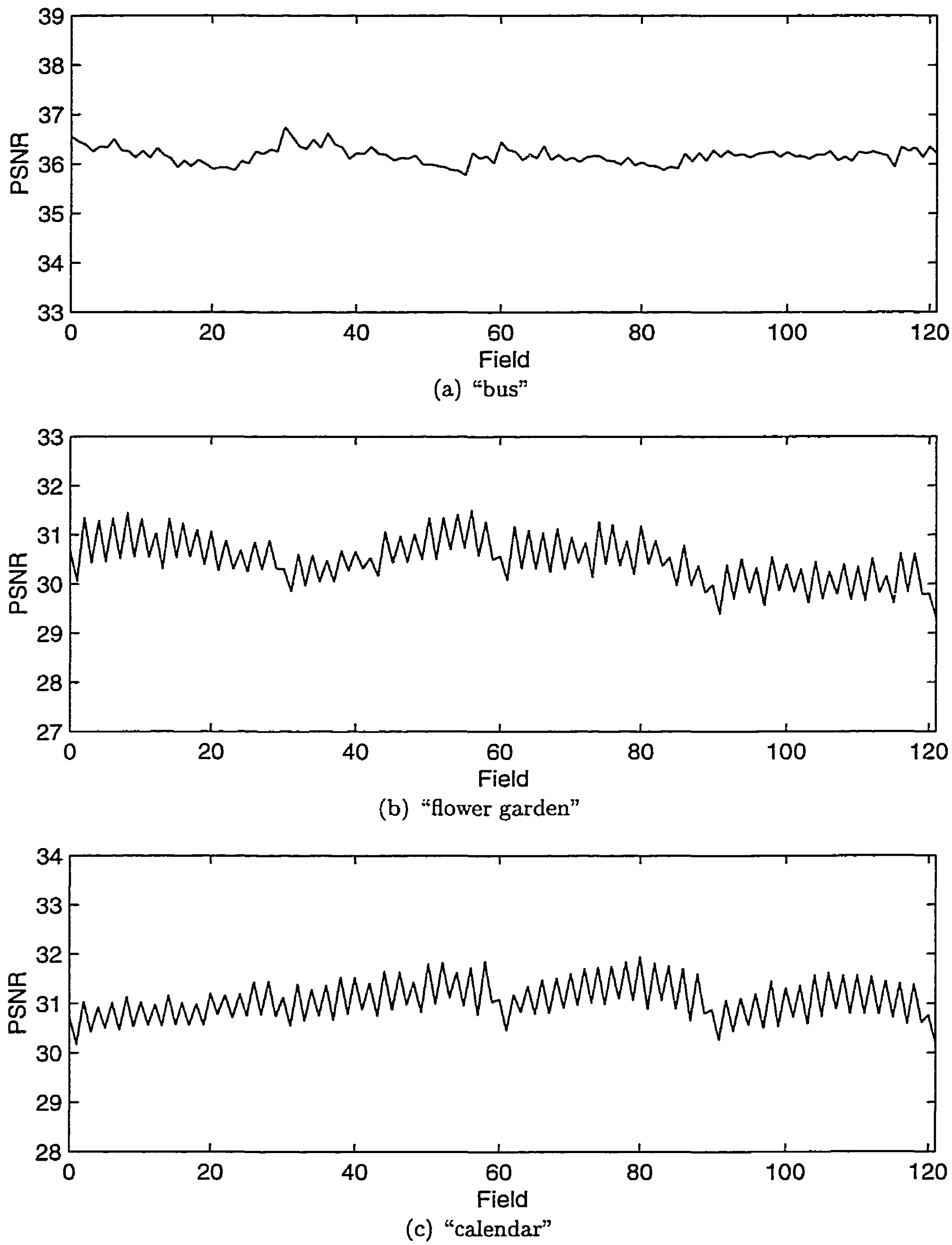
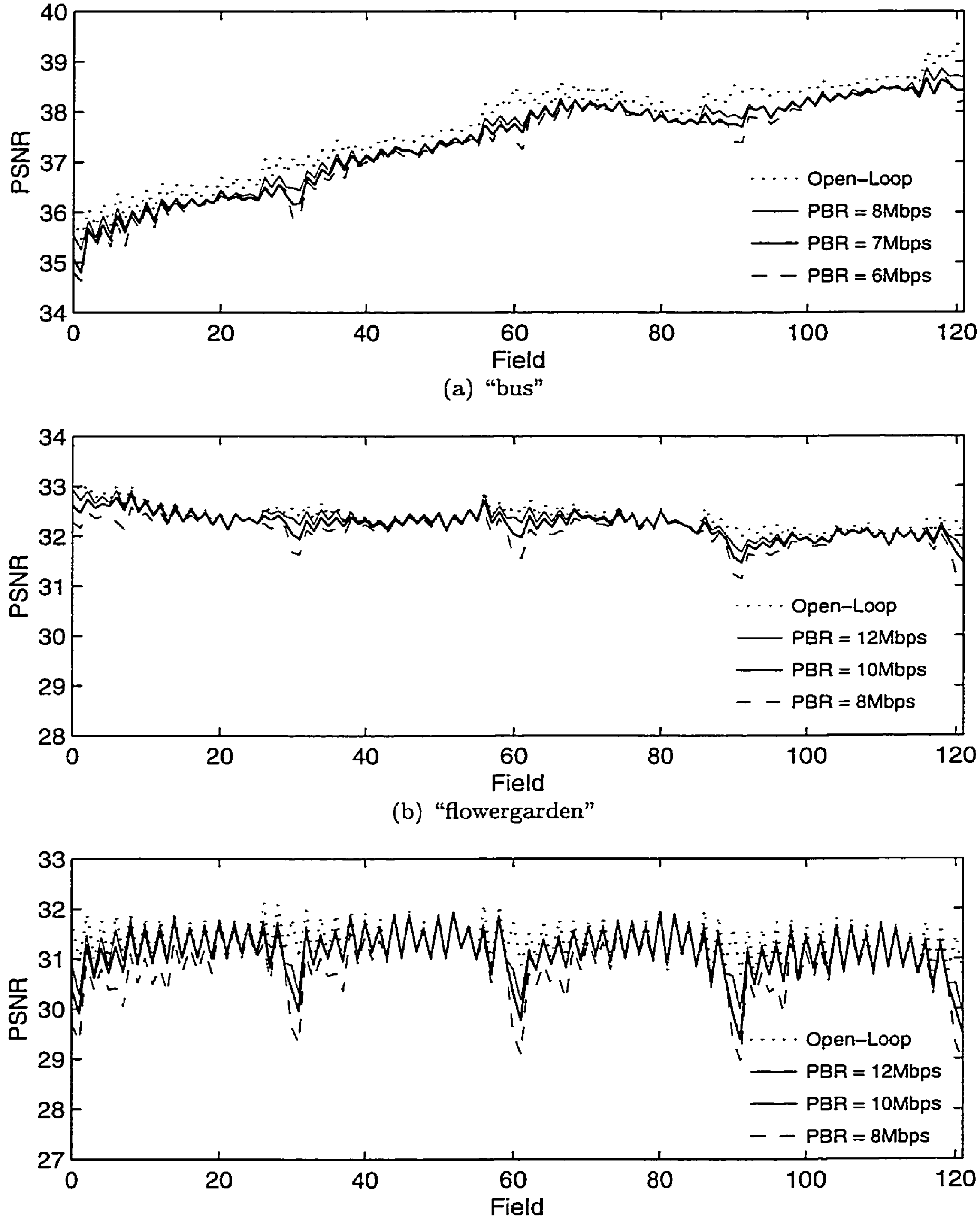


Fig. B.6 TM5 chrominance-2 PSNR curves for different values of Q

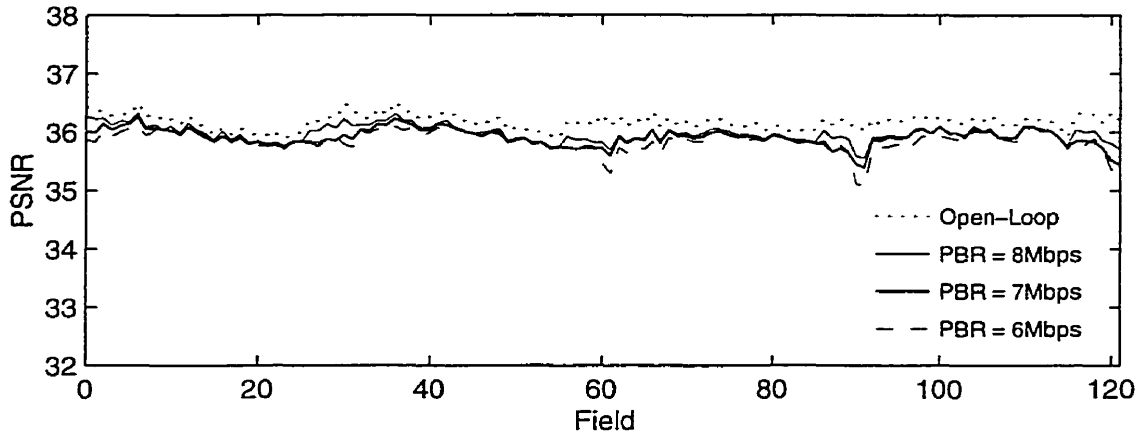


(a) "bus"

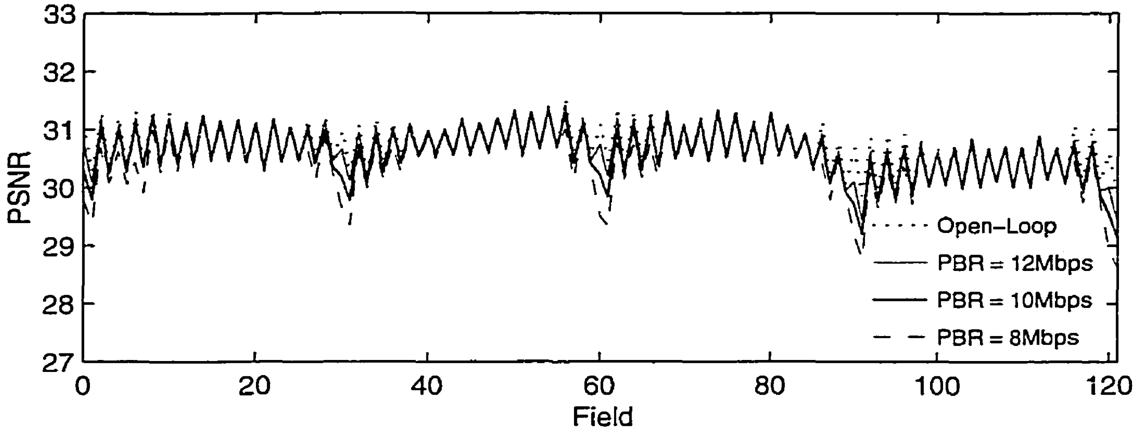
(b) "flowergarden"

(c) "calendar"

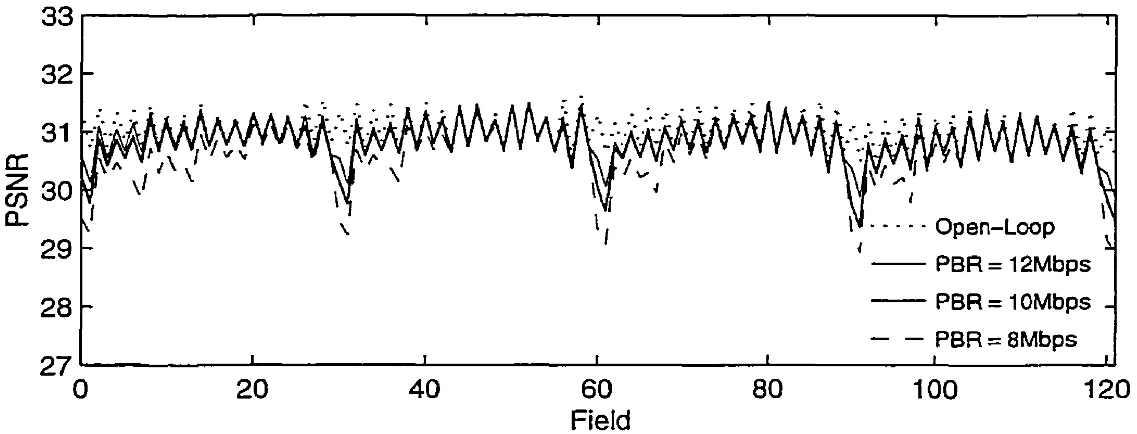
Fig. B.7 Chrominance-1 PSNR curves for PBR traffic shaping



(a) "bus"

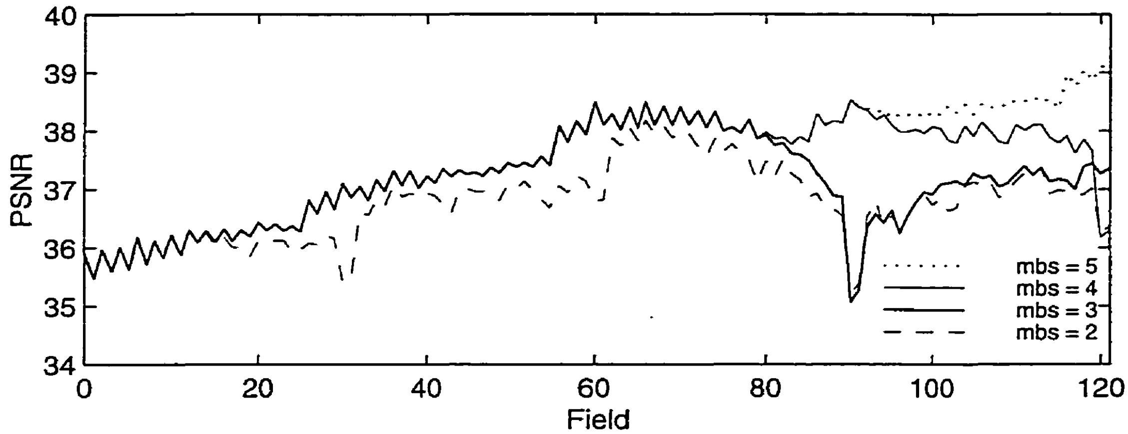


(b) "flowergarden"

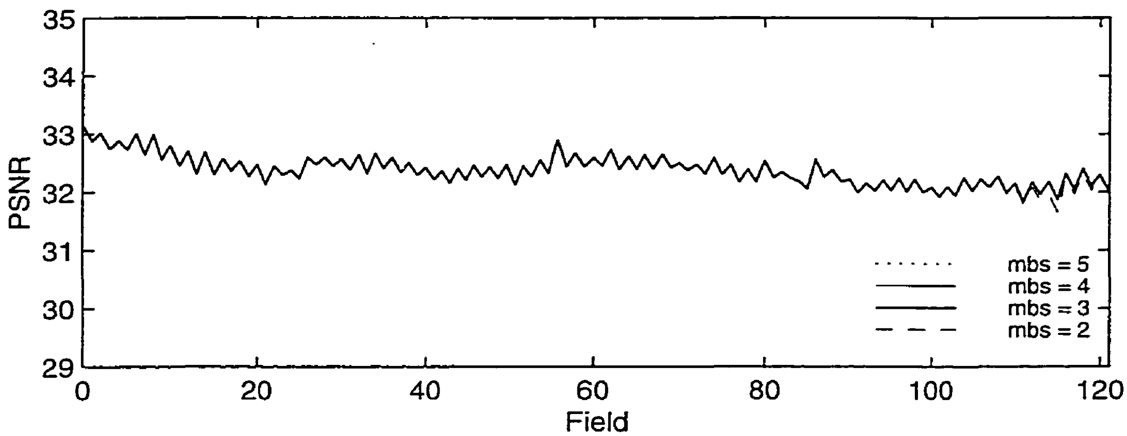


(c) "calendar"

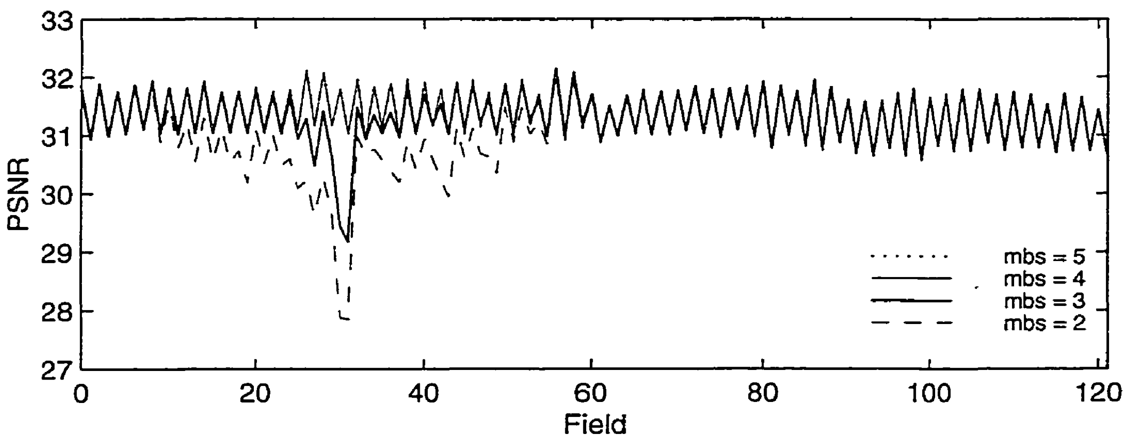
Fig. B.8 Chrominance-2 PSNR curves for PBR traffic shaping



(a) "bus"

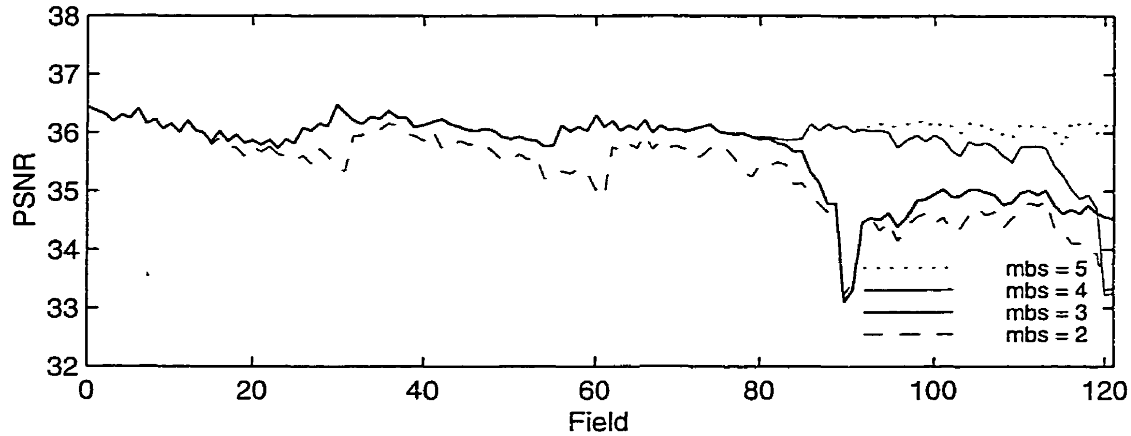


(b) "flowergarden"

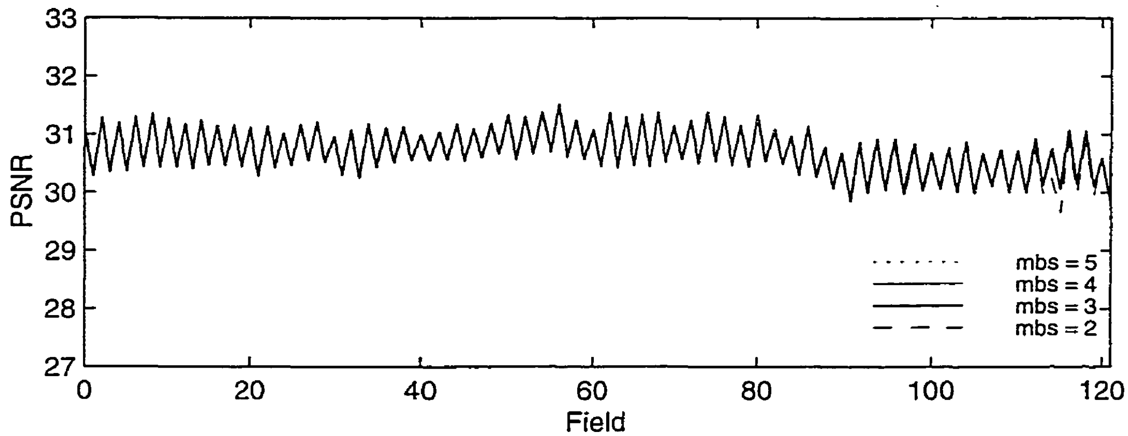


(c) "calendar"

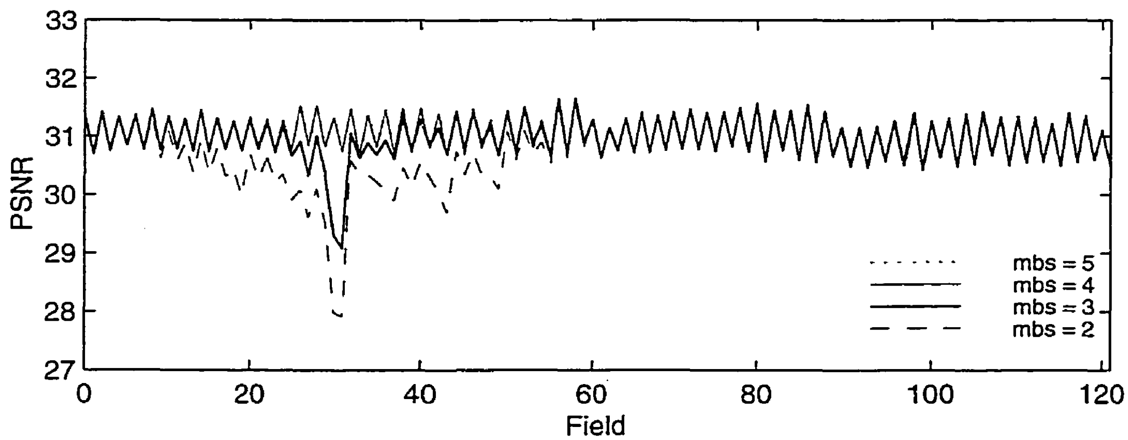
Fig. B.9 Chrominance-1 PSNR curves for SBR traffic shaping



(a) "bus"

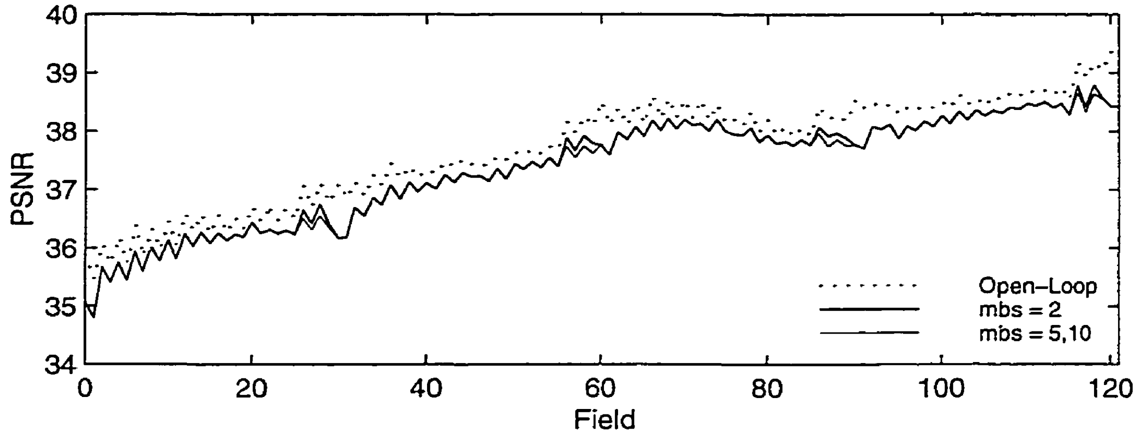


(b) "flowergarden"

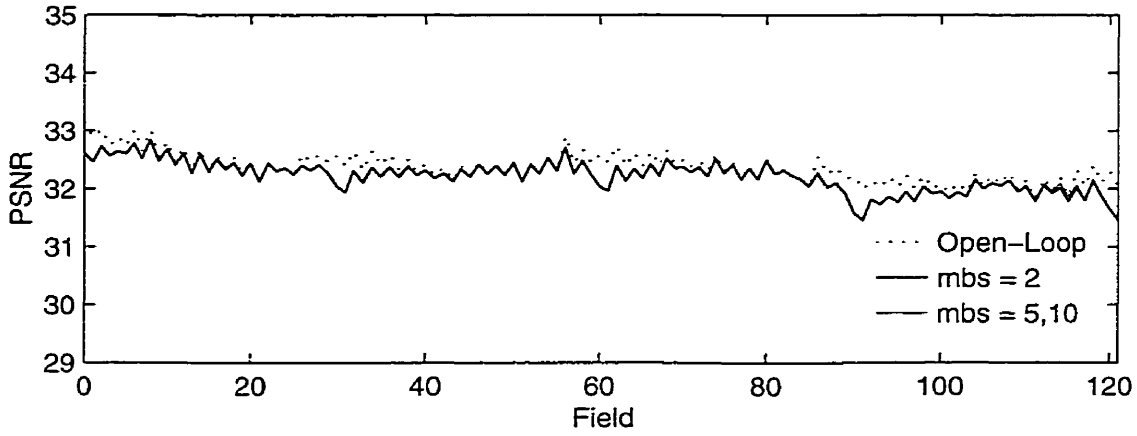


(c) "calendar"

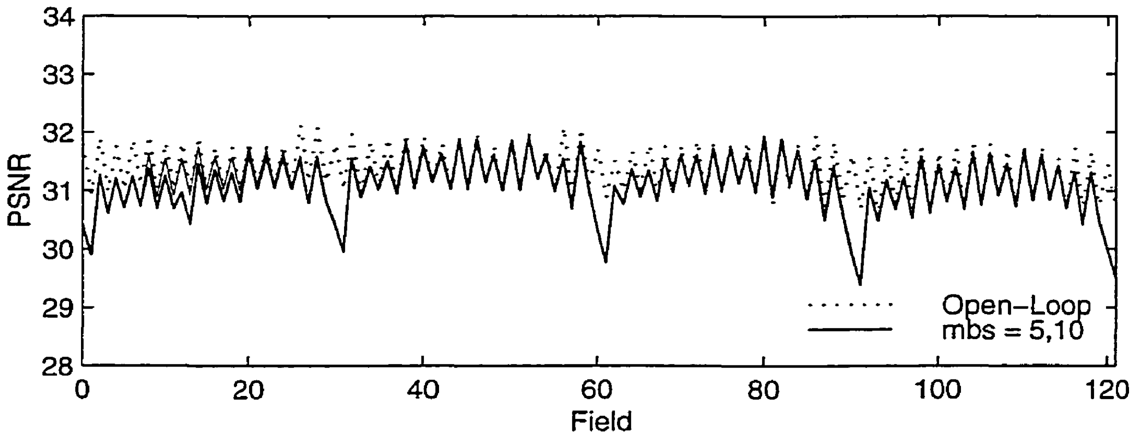
Fig. B.10 Chrominance-2 PSNR curves for SBR traffic shaping



(a) "bus"

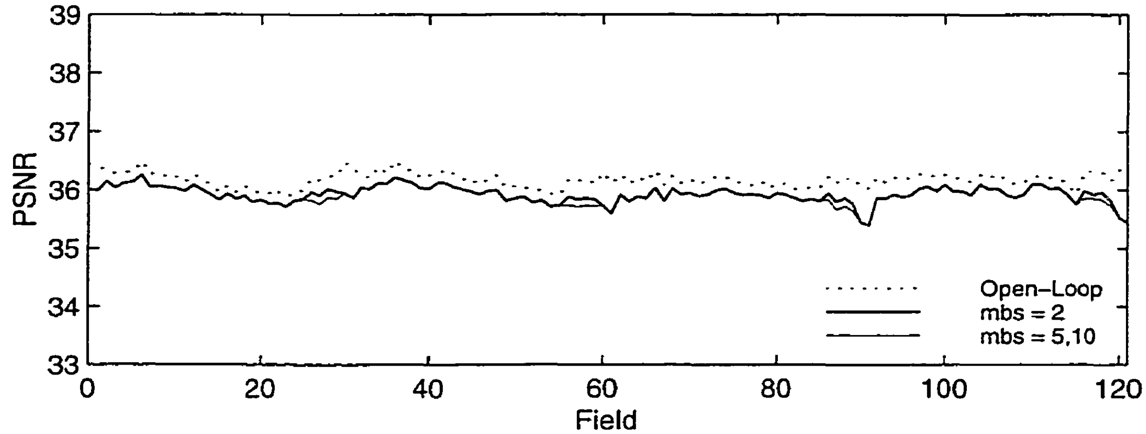


(b) "flowergarden"

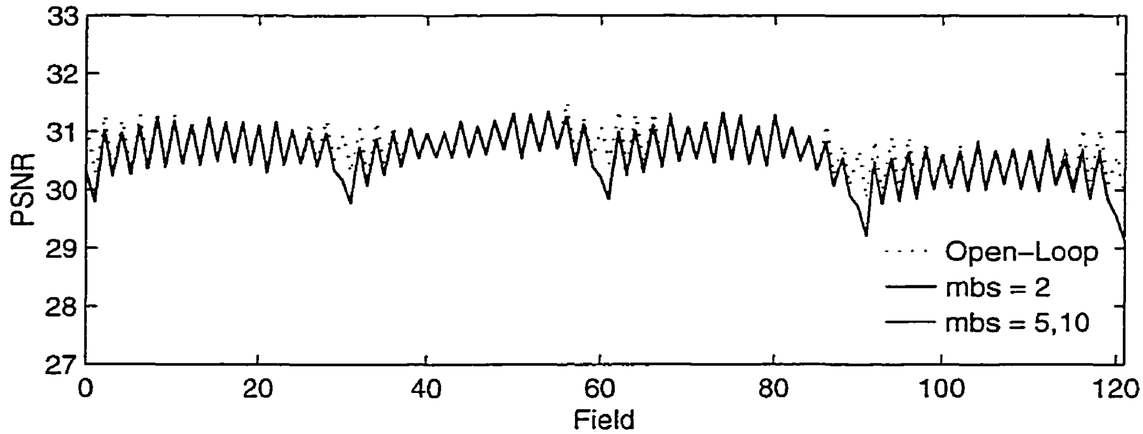


(c) "calendar"

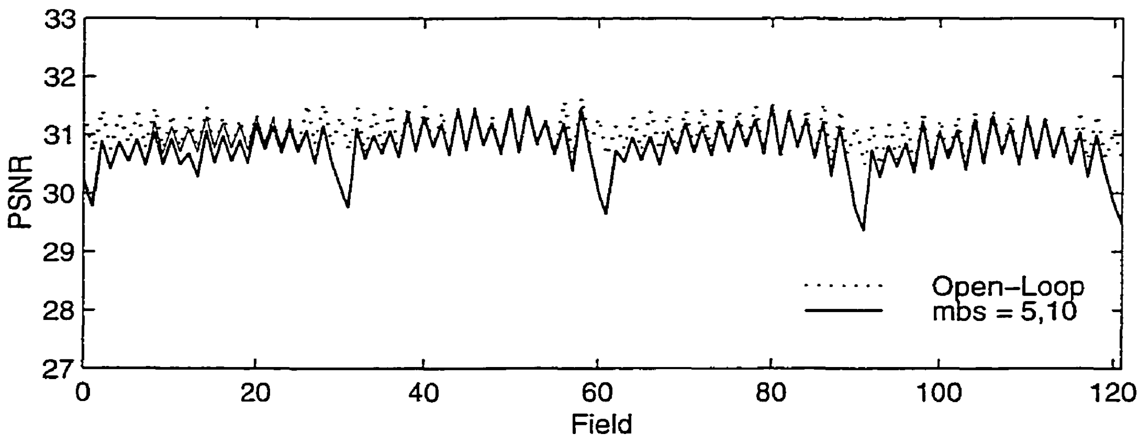
Fig. B.11 Chrominance-1 PSNR curves for MBS traffic shaping



(a) "bus"



(b) "flowergarden"



(c) "calendar"

Fig. B.12 Chrominance-2 PSNR curves for MBS traffic shaping

Appendix C

Chrominance Tables

This appendix contains the tabulated PSNR statistics of the chrominance components of the images considered in the experiments. The organization is as follows:

1. **Variable Bit Rate** (Chapter 3)

   Tables C.1 (page 112) to C.2 (page 112).

2. **Constant Bit Rate** (Chapter 3)

   Two sets of constant bit rate experimental results are provided — one for the results obtained through the use of the rate constrained coder and the other through the use of the Test Model 5 coder.

   - *Rate-Constrained Coder:* Tables C.3 (page 113) to C.4 (page 113).
   - *Test Model 5:* Tables C.5 (page 113) to C.6 (page 113).

3. **Traffic Shaping** (Chapter 4)

   Three different sets of experimental results involving traffic shaping are provided — peak bit rate limiting, sustained bit rate limiting, and maximum burst size adjusting.

   - *Peak-Bit-Rate Limiting:* (Section 4.4.1) Tables C.7 (page 114) to C.8 (page 114).

- *Sustained-Bit-Rate Limiting:* (Section 4.4.2) Tables C.9 (page 116) to C.10 (page 116).
- *Maximum-Burst-Size Adjusting:* (Section 4.4.3) Tables C.11 (page 117) to C.12 (page 117).

Sequence	*Q*	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	6	39.9164	38.2068	36.0501	0.8976
	8	39.3621	37.6924	35.4742	0.9278
	10	38.8857	37.0808	34.9833	0.9820
flower gar.	6	33.6529	32.9019	32.3346	0.2729
	8	33.1504	32.3774	31.8363	0.2589
	10	32.7751	31.9929	31.4674	0.2519
calendar	6	32.8951	32.0831	31.1049	0.5823
	8	32.1318	31.3768	30.5834	0.4571
	10	31.4810	30.7946	30.1155	0.3852

Table C.1 VBR coded chrominance-1 PSNR statistics

Sequence	*Q*	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	6	36.9154	36.5784	36.2878	0.1366
	8	36.5072	36.1737	35.8823	0.1291
	10	36.2176	35.7586	35.3377	0.1822
flower gar.	6	32.2481	31.3655	30.3030	0.5197
	8	31.5181	30.7212	29.8279	0.4327
	10	31.0011	30.2027	29.4063	0.3762
calendar	6	32.3386	31.6599	30.9465	0.4589
	8	31.6367	31.0338	30.4248	0.3680
	10	31.0896	30.5251	29.9749	0.3054

Table C.2 VBR coded chrominance-2 PSNR statistics

Sequence	*n*	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	1	38.2366	36.8672	34.1321	0.9637
	3	38.8857	37.0067	34.8043	1.0637
	5	39.1144	37.1511	34.9833	1.0579
flower gar.	1	32.3186	31.5476	30.3352	0.4161
	3	32.5138	31.8094	31.0870	0.3081
	5	32.7751	31.9681	31.2444	0.3208
calendar	1	31.3703	30.1235	27.6677	0.8116
	3	31.3705	30.2564	29.0384	0.5728
	5	31.7096	30.5930	29.4246	0.5821

Table C.3 CBR coded chrominance-1 PSNR statistics

Sequence	*n*	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	1	36.0525	35.5238	34.4844	0.3069
	3	36.3417	35.6045	35.0556	0.2628
	5	36.3417	35.7092	35.0499	0.2615
flower gar.	1	30.8389	29.7376	27.6895	0.6834
	3	30.8558	29.7641	28.7124	0.4451
	5	31.1555	30.0263	29.0088	0.4472
calendar	1	30.9550	29.8847	27.8071	0.7011
	3	31.0078	30.0239	28.8730	0.5073
	5	31.2918	30.3430	29.2859	0.4840

Table C.4 CBR coded chrominance-2 PSNR statistics

Sequence	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	39.2876	37.7069	35.6964	0.9026
flower gar.	33.1266	32.2824	31.6137	0.3289
calendar	32.3452	31.3836	30.3268	0.5112

Table C.5 TM5 coded chrominance-1 PSNR statistics

Sequence	*max(dB)*	*avg(dB)*	*min(dB)*	*std(dB)*
bus	36.7458	36.1664	35.7681	0.1685
flower gar.	31.5074	30.4864	29.2933	0.5024
calendar	31.9508	31.0601	30.1731	0.4262

Table C.6 TM5 coded chrominance-2 PSNR statistics

Traffic Descriptors			*Measured Statistics*			
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (dB)*	*avg (dB)*	*min (dB)*	*std (dB)*
Sequence: "bus"						
8.0000	3.8482	5	38.8595	37.4244	35.2419	0.9054
7.0000	3.8482	5	38.6556	37.3550	34.8043	0.9287
6.0000	3.8482	5	38.7571	37.2795	34.6268	0.9511
Sequence: "flower garden"						
12.0000	4.4462	5	32.9462	32.2969	31.6740	0.2680
10.0000	4.4462	5	32.8376	32.2236	31.4576	0.2663
8.0000	4.4462	5	32.5953	32.0966	31.0736	0.2910
Sequence: "calendar"						
12.0000	4.8884	5	31.9143	31.1972	29.8817	0.4728
10.0000	4.8884	5	31.9371	31.1192	29.3893	0.5289
8.0000	4.8884	5	31.8846	30.8554	28.9023	0.6357

Table C.7 Chrominance-1 PSNR statistics for PBR traffic shaping

Traffic Descriptors			*Measured Statistics*			
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (dB)*	*avg (dB)*	*min (dB)*	*std (dB)*
Sequence: "bus"						
8.0000	3.8482	5	36.3253	35.9743	35.5554	0.1409
7.0000	3.8482	5	36.2554	35.9061	35.3945	0.1550
6.0000	3.8482	5	36.1015	35.8475	35.0969	0.1782
Sequence: "flower garden"						
12.0000	4.4462	5	31.4312	30.6439	29.4063	0.4497
10.0000	4.4462	5	31.3419	30.5800	29.1183	0.4800
8.0000	4.4462	5	31.4093	30.4378	28.6039	0.5474
Sequence: "calendar"						
12.0000	4.8884	5	31.1957	30.8757	29.8264	0.3783
10.0000	4.8884	5	31.4958	30.7984	29.3677	0.4392
8.0000	4.8884	5	31.4670	30.5726	28.9017	0.5469

Table C.8 Chrominance-2 PSNR statistics for PBR traffic shaping

Traffic Descriptors			*Measured Statistics*			
R_P (Mbps)	R_S (Mbps)	*mbs* (frames)	*max* (dB)	*avg* (dB)	*min* (dB)	*std* (dB)
Sequence: "bus"						
10.0286	3.0000	2	38.1826	36.7568	35.1738	0.6602
10.0286	3.0000	3	38.4990	37.0877	35.0602	0.7682
10.0286	3.0000	4	38.5372	37.3929	35.4742	0.8335
10.0286	3.0000	5	39.1144	37.5498	35.4742	0.9399
Sequence: "flower garden"						
14.0688	4.0000	2	33.1504	32.3687	31.6501	0.2704
14.0688	4.0000	3	33.1504	32.3774	31.8363	0.2589
14.0688	4.0000	4	33.1504	32.3774	31.8363	0.2589
14.0688	4.0000	5	33.1504	32.3774	31.8363	0.2589
Sequence: "calendar"						
15.6427	4.0000	2	32.0168	31.0320	27.8457	0.6823
15.6427	4.0000	3	32.1475	31.2978	29.1825	0.5075
15.6427	4.0000	4	32.1318	31.3768	30.5834	0.4571
15.6427	4.0000	5	32.1318	31.3768	30.5834	0.4571

Table C.9 Chrominance-1 PSNR statistics for SBR traffic shaping

Traffic Descriptors			*Measured Statistics*			
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (dB)*	*avg (dB)*	*min (dB)*	*std (dB)*
Sequence: "bus"						
10.0286	3.0000	2	36.4404	35.3307	33.2374	0.7276
10.0286	3.0000	3	36.4799	35.6646	33.1137	0.6737
10.0286	3.0000	4	36.4799	35.9291	33.2296	0.4499
10.0286	3.0000	5	36.4799	36.0763	35.7378	0.1479
Sequence: "flower garden"						
14.0688	4.0000	2	31.5181	30.7038	29.6478	0.4454
14.0688	4.0000	3	31.5181	30.7212	29.8279	0.4327
14.0688	4.0000	4	31.5181	30.7212	29.8279	0.4327
14.0688	4.0000	5	31.5181	30.7212	29.8279	0.4327
Sequence: "calendar"						
15.6427	4.0000	2	31.5779	30.7346	27.9109	0.5982
15.6427	4.0000	3	31.6560	30.9669	29.0864	0.4241
15.6427	4.0000	4	31.6367	31.0338	30.4248	0.3680
15.6427	4.0000	5	31.6367	31.0338	30.4248	0.3680

Table C.10 Chrominance-2 PSNR statistics for SBR traffic shaping

Traffic Descriptors			*Measured Statistics*			
R_P *(Mbps)*	R_S *(Mbps)*	*mbs (frames)*	*max (dB)*	*avg (dB)*	*min (dB)*	*std (dB)*
Sequence: "bus"						
7.0000	3.8482	2	38.7890	37.3714	34.8043	0.9337
7.0000	3.8482	5	38.6556	37.3550	34.8043	0.9287
7.0000	3.8482	10	38.6556	37.3550	34.8043	0.9287
Sequence: "flower garden"						
10.0000	4.4462	2	32.8376	32.2204	31.4576	0.2695
10.0000	4.4462	5	32.8376	32.2236	31.4576	0.2663
10.0000	4.4462	10	32.8376	32.2236	31.4576	0.2663
Sequence: "calendar"						
10.0000	4.8884	2	31.9372	31.0855	29.3893	0.5228
10.0000	4.8884	5	31.9372	31.1192	29.3893	0.5289
10.0000	4.8884	10	31.9372	31.1192	29.3893	0.5289

Table C.11 Chrominance-1 PSNR statistics for different values of mbs

Traffic Descriptors			Measured Statistics			
R_P (Mbps)	R_S (Mbps)	mbs (frames)	max (dB)	avg (dB)	min (dB)	std (dB)
Sequence: "bus"						
7.0000	3.8482	2	36.2554	35.9223	35.3945	0.1459
7.0000	3.8482	5	36.2554	35.9061	35.3945	0.1550
7.0000	3.8482	10	36.2554	35.9061	35.3945	0.1550
Sequence: "flower garden"						
10.0000	4.4462	2	31.3036	30.3976	28.6039	0.5401
10.0000	4.4462	5	31.4093	30.4378	28.6039	0.5474
10.0000	4.4462	10	31.4093	30.4378	28.6039	0.5474
Sequence: "calendar"						
10.0000	4.8884	2	31.4958	30.7675	29.3677	0.4342
10.0000	4.8884	5	31.4958	30.7984	29.3677	0.4392
10.0000	4.8884	10	31.4958	30.7984	29.3677	0.4392

Table C.12 Chrominance-2 PSNR statistics for different values of mbs

Bibliography

[1] N. Ohta, *Packet Video: Modeling and Signal Processing.* Artech House Inc., 1994.

[2] C. A. Poynton, *A Technical Introduction to Digital Video.* John Wiley & Sons Inc., 1996.

[3] A. Puri, "Video coding using the MPEG-2 compression standard," *Proceedings of SPIE*, vol. 2094, pp. 1701–1713, 1993.

[4] D. J. Le Gall, "The MPEG video compression algorithm," *Signal Processing: Image Communication*, vol. 4, pp. 129–140, 1992.

[5] P. N. Tudor, "MPEG-2 video compression," *Electronics & Communication Engineering Journal*, pp. 257–264, Dec. 1995.

[6] E. Dubois, "Source compression – video," in *The Communications Handbook* (J. Gibson, ed.), ch. 102, pp. 1449–1461, CRC Press, 1997.

[7] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression.* New York: Plenum Press, 1991.

[8] J. P. Leduc and C. Labit, "Digital television on ATM networks: How to optimize coding and transmission?," *Proceedings of SPIE*, pp. 1044–1055, 1995.

[9] B. Girod, "Scalable video for multimedia workstations," *Comput. and Graphics*, vol. 17, no. 3, pp. 269–276, 1993.

[10] International Standard ISO/IEC 13818-2, *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video*, 1995.

[11] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 software video codec," *Proceedings of SPIE*, vol. 2419, pp. 100–108, Feb. 1995. San Jose, California.

[12] D. Reininger, G. Ramamurthy, and D. Raychaudhuri, "VBR MPEG video coding with dynamic bandwidth renegotiation," in *1995 IEEE International Conference on Communications* [37], pp. 1773–1777. Seattle, Washington.

[13] "MPEG-2 FAQ." http://www.crs4.it./ luigi/MPEG/.

[14] A. L. N. Reddy and R. Haskin, "Video servers," in *The Communications Handbook* (J. Gibson, ed.), ch. 106, pp. 1502–1515, CRC Press, 1997.

[15] G. Neufeld, D. Makaroff, and N. Hutchinson, "The design of a variable bit rate continous media server," in *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video* [36], pp. 375–378. Durham, New Hampshire.

[16] W. Verbiest, L. Pinnoo, and B. Voeten, "The impact of the ATM concept on video coding," *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 1623–1632, Dec. 1988.

[17] D. Raychaudhuri, D. Reininger, M. Ott, and G. Welling, "Multimedia processing and transport for the wireless terminal scenario," *Proceedings of SPIE*, vol. 2501, pp. 1056–1067, 1995.

[18] A. Ortega, M. W. Garrett, and M. Vetterli, "Rate constraints for video transmission over ATM networks based on joint source/network criteria," *Annales des Télécommunications*, vol. 50, no. 7–8, pp. 603–616, 1995.

[19] H. Heeke, "A traffic-control algorithm for ATM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 182–189, June 1993.

[20] G. Karlsson, "Asynchronous transfer of video," tech. rep., Swedish Institute of Computer Science (SICS), 1995.

[21] G. Karlsson, "Asynchronous transfer of video," *IEEE Communications Magazine*, vol. 34, pp. 118–126, Aug. 1996.

[22] Y.-Q. Zhang *et al.*, "Variable bit-rate video transmission in the broadband ISDN environment," *Proceedings of the IEEE*, vol. 79, pp. 214–221, Feb. 1991.

[23] R. O. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues.* Artech House Inc., 1994.

[24] D. Bertsekas and R. Gallager, *Data Networks.* Prentice Hall, second ed., 1992.

[25] The ATM Forum, *ATM User-Network Interface Specification. version 3.1.* Prentice Hall, Sept. 1994.

[26] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the "leaky bucket" policing mechanism in ATM networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 335–342, Apr. 1991.

[27] I. E. G. Richardson and M. J. Riley, "Usage parameter control cell loss effects on MPEG video," in *1995 IEEE International Conference on Communications* [37], pp. 970–974. Seattle, Washington.

[28] M. R. Izquierdo and D. R. Reeves, "Statistical characterization of MPEG VBR video at the slice layer," *Proceedings of SPIE*, vol. 2417, pp. 268–279, 1995.

[29] ISO/IEC JTC1/SC29/WG11/N0400, *Test Model 5*, 1993. Version 2.

[30] A. Eleftheriadis and D. Anastassiou, "Meeting arbitrary QoS constraints using dynamic rate shaping coding digital video," in *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video* [36], pp. 95–106. Durham, New Hampshire.

[31] H. Zhang and E. W. Knightly, "A new approach to support delay-sensitive VBR video in packet-switched networks," in *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video* [36], pp. 275–286. Durham, New Hampshire.

[32] M. Kawashima *et al.*, "Adaptation of the MPEG video-coding algorithm to network applications," *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 3, pp. 261–269, Aug. 1993.

[33] T. Han and L. Orozco-Barbosa, "Performance requirements for the transport of MPEG video streams over ATM networks," in *1995 IEEE International Conference on Communications* [37], pp. 221–225. Seattle, Washington.

[34] P. Pancha and M. El Zarki, "MPEG coding for variable bit rate video transmission," *IEEE Communications Magazine*, pp. 54–66, May 1994.

[35] M. Ghanbari and J. Azari, "Effect of bit rate variation of the base layer on the performance of two-layer video codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 1, pp. 8–17, 1994.

[36] IEEE Communications Society, *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, Apr. 1995. Durham, New Hampshire.

[37] IEEE, *1995 IEEE International Conference on Communications*, June 1995. Seattle, Washington.

IMAGE EVALUATION TEST TARGET (QA-3)

150mm

6″

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989