COLOR PROCESSING AND RATE CONTROL FOR STORAGE AND TRANSMISSION OF DIGITAL IMAGE AND VIDEO

by

Sang-Yong Lee

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

May 2003

Dedication

To My Parents.

Acknowledgements

First of all, I would like to express my gratitude to my advisor Professor Antonio Ortega for his support and help. His insights always made my vague ideas clear and meaningful. I would also like to thank him for allowing me to focus on research.

I would like to thank Professor C.-C.Jay Kuo for giving me lots of advice and encouragement and for serving on my dissertation committee, Professor Zhen Zhang for enlightening me through his four valuable lectures, Professor Cyrus Shahabi for serving on my dissertation committee, Professor Daniel C. Lee, Professor Shrikanth S. Narayanan, Professor Ashish Goel for serving on my qualifying committee.

I would also like to thank my group mates including Youngjun who helped me join this wonderful group, Woontack, Wenqing, Krisda, Raghavendra, Younggap and Paul who helped me adapt to the group, and Hyungsuk, Phoom, Baltasar, Miao, Wendi and Hua who shared lots of their ideas with me. In particular, I

would like to thank Hyukjun and Naveen who helped me a lot including reviewing parts of my paper.

I also want to take this opportunity to thank lots of KESO members and their families including Jietae Shin, Yongdae, Dongjoon, Jingyeong, Younggook, Chulmin, Changki, Kitae, Soonil, Changsung and Janghoon.

I thank my wife for her love, sacrifice and help. I remember having the happiest moments with her and my son. I want to thank my mother and my brother and sister for their endless love and support. My deepest gratitude goes to my father whom I really miss in my heart.

Contents

List of Tables

List of Figures

Abstract

Multimedia technology has taken an increasingly important part in our daily life. In particular, fast development of technology now enables handling and transmission of high volume data like images and videos. Although the cost of storage and channel bandwidth is becoming lower, it is still too high to handle multimedia data without compression. Therefore we need to select a proper compression ratio so as to avoid any applicable constraints, while providing high perceptual quality. The applications considered in this thesis are digital cameras and video-on-demand (VOD) services.

Most digital cameras have limited storage size and time delay for compression due to the limited working memory. We address the problem of image compression in digital cameras, where the goal is to achieve better quality at a given rate. A novel method including format conversion, nonlinear transform and cropping is proposed to compress captured image data without increasing redundancy.

We also consider the problem of online rate control in digital cameras, where the goal is to achieve near-constant distortion for each image. An online rate control algorithm based on the amount of storage used by previously stored images, the current received image, and the estimated rate of future images is proposed. A MMAX (minimization of maximum distortion) criterion is used, since each image has the same importance.

VOD services use either a constant bit rate (CBR) or variable bit rate (VBR) channel to transmit video streams. In these systems, the main constraints are the maximum end-to-end delay to support real-time playback and the channel rates. We focus on finding an off-line optimal rate control in both CBR and VBR transmission. To provide best minimum quality to each data unit, a MMAX criterion is used. We introduce an approach that minimizes the average distortion using the leftover bit-budget available after the MMAX solution (MMAX+). This allows us to increase the overall quality, and can also reduce the complexity of MMSE bit allocation relative to searching for the optimal MMSE solution directly.

Chapter 1

Introduction

1.1 Motivation of the research

The fast development of technology enables the increasing use of digital media in commercial products. The advantage of digital formats is that they give many desirable features such as flexibility and robustness. Also the development of the Internet enables the easy exchange of digital data, including images and videos. New image and video input products such as digital cameras and camcorders are becoming available.

In commercial video input devices, digital technology was first introduced to improve camera parts in analog camcorders. By using a single DSP processor, many data processing tasks, such as interpolation, gamma correction, exposure control and white-balance control can be implemented efficiently. But in analog camcorders, digitally processed data are converted into an analog signal by using

D/A converters, re-formatted to television signal such as NTSC, PAL or SECAM format and modulated to be stored on magnetic tapes [30]. Therefore the main purpose of digital signal processing in these systems is making the CCD input source as similar as possible to the original scene. More recently, as a descendant of this analog camcorder, digital cameras (including digital camcorders) are becoming very popular. The main difference between digital cameras and analog camcorders with a DSP processor is that the output of the digital camera is stored in digital format. Therefore stored data can be copied without additional distortion and can be edited easily. But the price of digital storage, e.g., flash memory devices, is still very high, and therefore the bit-rate of the images should be kept as low as possible. To do this, most digital cameras use lossy compression schemes to store the images. Therefore most digital cameras use image processing techniques to improve image quality, such as those first used in analog camcorders, and then compress the image as much as possible while preserving perceptual quality (see Fig. 1.1). In other words, the image processing and image compression stages are totally separated. Here we argue that this approach may not be the best in the sense of maximizing the quality of decoded images under a given bit-budget. In this thesis, we thus study techniques to merge the image processing and image compression stages in order to improve the quality of stored images.

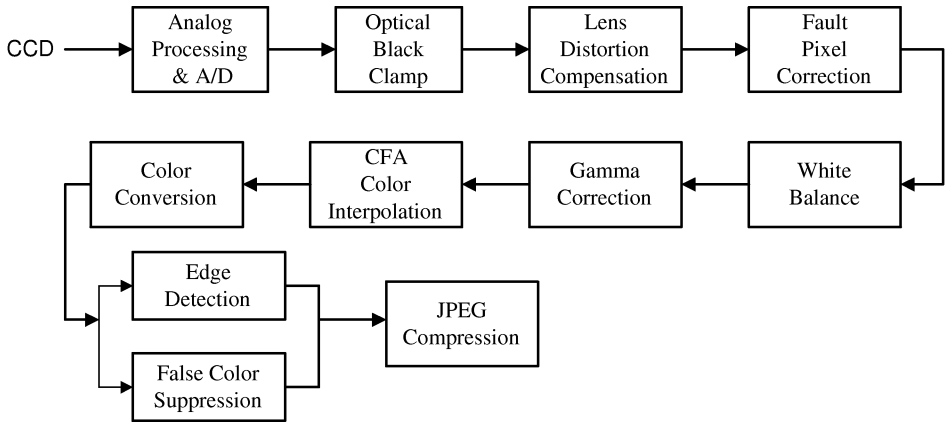


Figure 1.1: The block diagram of image processing and image compression units in digital cameras. [24]

Another problem related to data compression in digital input devices is that of sharing the available and possibly scarce storage among several images or videos. Efficient source coding is indispensable to store images and videos in a compressed form. Many data compression formats have been standardized in order to facilitate interoperability. In particular, lossy coding techniques like JPEG [49], JPEG2000 [66] and JBIG [22] for images, and MPEGx [25, 26, 27], H.26x [29, 77] for videos are generally used to achieve high compression ratio. All of these standards use entropy coding to reduce the redundancy in the data. However, the amount of redundancy varies within each source, e.g., within a video sequence or from image to image, and, as a result, the bit-rate of encoded data is variable. Although each image or each video frame can have near-constant bit-rate by choosing different quantization values, the output rate should be variable

in order to achieve desirable features like minimum average distortion, constant distortion or constant visual quality.

To store variable rate data into a fixed size storage, proper bit allocation techniques are needed. For example, in digital cameras, an "on-line" bit allocation method is needed since no information about the statistics of future images is available when the current image is being compressed. One simple method is to allocate the same bit-budget to each image, but in this case, the visual quality of each coded image can fluctuate significantly . Therefore, to achieve equal visual quality of all images, the statistics of future images should be estimated. In other scenarios, if the goal is to transfer some multimedia data from source storage media to another storage medium by using a detachable small size storage media like a floppy disk or flash memory, all the statistics of the source to transfer are given before starting data compression. So in this "off-line" bit allocation problem, the optimal solution can be found under a given objective function.

In this thesis, we use a minimization of maximum distortion (MMAX) criterion to find the optimal solution since each data unit (image) is equally important (criteria for quality measurement will be mentioned in the next section). We also propose an "on-line" bit allocation algorithm to achieve a near-optimal solution under this criterion.

The last topic of this thesis is bit allocation for data transmission. Recently, high bandwidth video applications over networks are becoming popular, and include for example video conferencing and video-on-demand (VOD). As in the other cases discussed above, "lossy compression" is used and so the output data rate of the coder tends to be variable bit rate (VBR). This rate is controlled by the encoder based on objectives such as coded video quality or data rate. Also, video transmission needs to be performed under delay constraints for real time playback since late video frames are useless.

To transmit VBR data, VBR transmission is better than CBR (constant bit rate) transmission, since VBR transmission needs lower end-to-end delay and a smaller buffer size [5]. Asynchronous Transfer Mode (ATM) networks are an example of a network architecture that allows VBR transmission with QoS (quality of service) guarantees, where the parameters specified to define QoS can be delay jitter, bandwidth, end-to-end delay and so on. Due to the limited network resources, negotiation between each user and the network is indispensable in order to ensure QoS guarantees. In addition, policing mechanisms are used to alert the network about users who violate the agreed upon transmision parameters. VBR video transmission through ATM networks has been studied with a leaky bucket policing function [53, 46, 5, 23]. In [53], VBR transmission under encoding and decoding buffer constraints and channel constraints is studied. In [46], rate control with multiple leaky bucket policing is introduced to regulate peak

rate. It may be desirable to supplement a traffic policing policy with a traffic shaping policy, where traffic shaping is used to smooth out a traffic flow. One simple traffic shaping approach is token bucket. In contrast to the leaky bucket, token bucket controls the flow of compliant cells [68]. In the encoder side, a leaky bucket and a token bucket are equivalent under the condition that the size of bucket is same and leak rate and token rate are same. But in the decoder side, the incoming data rate can be different.

The token bucket is also specified in next generation Internet Protocol (IP) networks. The Internet Engineering Task Force (IETF) has defined a Guaranteed Service (GS) in order to provide QoS to real time applications. As in the ATM case, token bucket policing is used as a traffic shaping method for a Guaranteed Service [62].

From the media providers view point, the goal is to supply the best quality videos, while using limited network bandwidth. In other words, the problem is how to allocate the bit-budget among several data units without violating the negotiated constraints and achieving best quality. In this thesis, we propose an off-line algorithm to find the optimal solution for this problem. The MMAX and additional minimization of average distortion (MMAX+) criteria are used to achieve the best minimum quality for each data unit and good average quality for a data sequence.

1.1.1 Focus of this thesis

This thesis presents topics related to lossy source coding optimization in several special environments. In Chapter 2 and Chapter 3, source coding problems for image (or video) input devices are studied. First, an algorithm to reduce redundancy by using the characteristics of the Bayer color CCD array is proposed and then an on-line bit allocation algorithm under a budget constraint is studied. In Chapter 4, the optimal bit allocation problem under channel constraints is studied. Token bucket and peak transmission rate are used as channel constraints. In this thesis, we focus especially on a minimization of maximum distortion criterion to find the optimal bit allocation.

The rest of this chapter will review bit allocation problems under different criteria. After that, the contributions on each topic of the thesis are summarized.

1.2 Bit Allocation

In this section, we review the bit allocation problems under several different constraints such as time delay, total bit-budget, and buffer and channel constraints. We focus on the case where each data unit has a finite set of R-D operating points (i.e., rate and distortion values of the coded data unit) determined by different quantization levels. Therefore the optimal bit allocation is the one that chooses a quantization level for each data unit, such that the corresponding rate

and distortion do not violate the given constraints and the desired cost function is minimized. To find the optimal bit allocation, at first, a distortion measure related to the desired goal should be defined.

1.2.1 Distortion measure

The most widely used distortion measure criterion in the field of multimedia data compression is the MMSE criterion. Under a given set of constraints, a MMSE solution uses the given bit-budget to decrease total distortion of the data sequence. This criterion, however, is not desirable in cases where constant distortion (or quality) is required. One example arises in the case of digital cameras, for which it is desirable that all images be stored with the same quality.

For this purpose, a MMAX criterion can be used. In this criterion, any remaining bit-budget is always assigned to the data unit that has maximum distortion. Therefore if there is a sufficiently fine granularity in the operating points for each data unit, it gives constant distortion to each data unit. Since the MMAX criterion is only minimizing maximum distortion, if the maximum distortion cannot be reduced then quality cannot be improved even though additional resources are available. The toy example in Fig. 1.2 illustrates this problem. After changing the quantization levels of both images, the distortion of the first image is 50 and that of the second image is 100, and the remaining bit-budget is 1KB. Using

1st image

r (KB)	d
2	110
4	50
5	40

2nd image

r (KB)	d
3	120
5	100
7	80

Figure 1.2: Tables indicate sets of available rate distortion pairs of two images. Total bit-budget is 10KB and the unit of rate in the tables is KB. The numbers in gray boxes indicate the rates corresponding to the MMAX solution.

the MMAX criterion, we try to decrease distortion of the second image using the remaining bit-budget. But going to the next finer quantization level would require a 2KB rate increase for the second image and a 1KB rate increase for the first image. In this case, we cannot reduce the maximum distortion because that would mean using an additional 2KB for the second image. Note that while using an additional 1KB for the first image would reduce the overall distortion it would not decrease the maximum distortion, and thus a MMAX search algorithm would not make that selection. Therefore additional distortion criteria are needed to improve the overall quality. The minimization of distortion under lexicographical constraints (MLEX) criterion and the MMAX+ (minimization of average distortion with the leftover bit-budget after the MMAX solution) criterion, which is introduced in this thesis, are used for this purpose.

In the MLEX criterion, two different solutions are arranged by a sorted list of their distortion in a non-increasing distortion order. Then a comparison of

the distortions is based on considering the list starting from the 1^{st} index. If the distortions in the first position are equal then the 2^{nd} indices are compared. If the distortion of two solutions is different for a given index then the solution with smaller distortion in that index is the better one. Otherwise the comparison is continued through the following position until the distortion of two solutions are different. [18, 19, 20, 21]. Therefore, this criterion focuses on improving the quality of each data unit. On the contrary, in a MMAX+ criterion, the remaining bit-budget after finding a MMAX solution is used to improve overall quality.

In this thesis, we mainly focus on MMAX and MMAX+ criteria to find an optimal solution.

1.2.2 Off-line algorithms vs. On-line algorithms

To find an optimal solution in a given criterion, all source information should be available before the bit-rate of each data unit is selected. However, in many cases, a current data unit should be coded without any knowledge of the statistics of future data units. In the former case an off-line method can be used, while in the latter on-line methods have to be used. In general applications, time delay, memory space and computing power are strongly limited, so an off-line method is usually not practical. But since the off-line method can give an optimal

solution, this method is often used to benchmark the performance of a given on-line method.

In some applications, such as developing DVD titles, all source data are available before coding, but due to limited computing power or time, gathering all statistics needed is impossible. In this case, partial information (including motion vectors and scene change locations) about the data is gathered in a first coding (called the first pass coding) and by using this information, a near optimal solution can be achieved in the second coding (called the second pass coding). These types of off-line methods are called 2-pass coding [79].

Window methods (such as jumping window and sliding window techniques) are also used as near optimal approaches in order to reduce complexity. In the methods, using the statistics of a limited number (the size of the window) of current and future data, the bit-rate of a current data unit is determined. In a jumping window method, the bit-rate of each data unit in a window is determined by using the statistics of the data units in the window. In a sliding window method, only the bit-rate of the current data unit is determined at each iteration and the window is shifted to determine the bit-rate of the next data unit.

An on-line method requires good estimation of the statistics of future data in order to achieve near optimal performance. Usually training data and previously coded data are used to estimate future data.

Chapter 3 of this thesis studies on-line bit allocation in a fixed size storage and Chapter 4 studies off-line optimal bit allocation for video transmission through CBR and VBR channels.

1.2.3 Independent allocation vs. Dependent allocation

In a source coding algorithm, entropy coding, in the form of Huffman coding or arithmetic coding, is normally used [8] [57]. For example, in JPEG coding, a modified Huffman coding is used in the base-line version and arithmetic coding is used as an option. Contrary to the quantization step in which data is compressed by removing less important information, data is compressed by removing redundant information in the entropy coding step. In a sequence of randomly selected images, such as those that could be captured by a digital camera, consecutive images seldom have high (temporal) correlation with each other, although there is high spatial correlation within each image. So in general, each image is coded separately. On the other hand, in a video sequence, consecutive frames have high temporal correlation. Therefore, after a frame has been coded by exploiting spatial correlation (called intra-frame coding), several following frames can be coded by exploiting the temporal correlation between the intra-coded frame and the new frames to be coded (called inter-frame coding). For instance, MPEG uses a group-of-pictures (GOP) layer and in the GOP (normally 12 or 15 frames), only

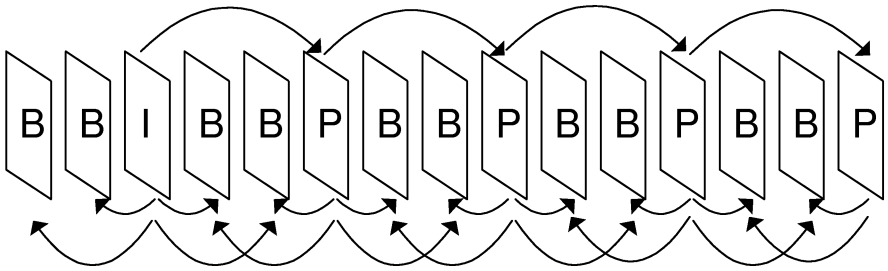


Figure 1.3: Typical structure of a GOP. I frame indicates intra-coded frame. P and B frames indicate inter-coded frames. The arrows indicate referring relation.

one frame is coded by using intra-frame coding and the remaining frames are coded by using inter-frame coding (see Fig. 1.3).

So, if each data unit is coded by only using its own correlation (intra-coding) then the distortion of one data unit does not affect the distortion of other data units. But if each data unit is coded by using the prediction based on the other data units and residues (inter-coding) then the distortion variation of the reference data unit affects the distortion of referring data units. In this case, the bit allocation problem is called a "dependent allocation", while otherwise it is called an "independent allocation". Dependent allocation is much more complex since rate and distortion of a referring data unit should be recalculated if the distortion of a reference data unit is changed.

1.2.4 Bit allocation under a total bit-budget constraint

This problem arises when the goal is to store data into fixed size of storage media. For example, a digital camera has a fixed memory size, which should be able to store a pre-determined number of shots. Another example is digital camcorders which store a pre-determined number of video frames into the fixed size tape media. This problem can be formulated as follows.

Under the MMSE criterion, the goal is to find the set of quantizers such that

$$J = \min_{q_i}(\sum_{i=1}^{N} d_i(q_i)) \quad \text{s.t.} \quad \sum_{i=1}^{N} r_i(q_i) \leq R \tag{1.1}$$

and using the MMAX criterion,

$$J = \min_{q_i}(\max_{i=1,\ldots,N}(d_i(q_i))) \quad \text{s.t.} \quad \sum_{i=1}^{N} r_i(q_i) \leq R \ , \tag{1.2}$$

where J is the minimum cost, N is the total number of data units, R is total bit-budget and r_i, d_i are rate and distortion of the i^{th} data unit, respectively. r_i and d_i are a function of q_i, where q_i is the quantization index of the i^{th} data unit. The range of q_i is from 1 to the number of quantization indices of the i^{th} data unit (Q_i).

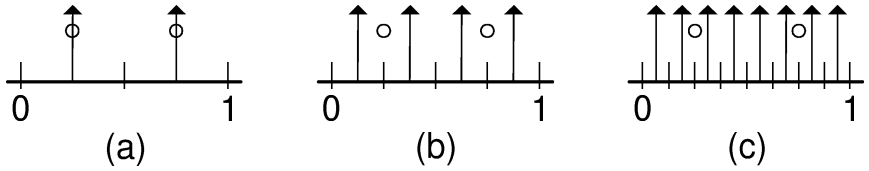


Figure 1.4: Toy example of increasing rate-distortion relation. x-axis indicates data range and arrow lines indicate reconstruction points. We assume the source data are 0.25 and 0.75. Rate and distortion (r,d) in (a),(b) and (c) are (1,0), (2,1/32) and (4,1/128) respectively. Here the square sum of error is used as a distortion. From (a) and (b), we can see that distortion is increased even if rate is increased. This phenomenon no longer occurs if we consider a large enough number of data inputs and the data is well distributed.

1.2.4.1 Optimal solution under the MMSE criterion

In the literature the Lagrangian method [64] and the dynamic programming method [48] are used as popular approaches to find the optimal solution under the MMSE criterion.

The Lagrangian approach to solve this problem is studied in [64]. Similarly, in [52], the Lagrangian method is used to find best wavelet bases to minimize distortion under given bit-budget. Also dependent allocation problem is studied in [51]. This dependent problem is also studied in [76], a model is defined and the optimal solution under MMSE and MMAX criteria is provided based on the model.

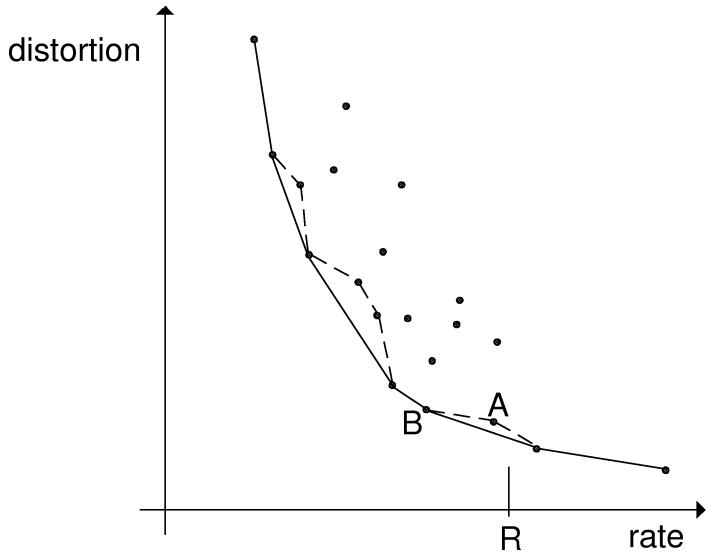


Figure 1.5: The curve indicates the convex hull and the points on the solid and dashed lines are the point of the operational rate distortion curve. Point A indicates the optimal solution of given total budget (R) and point B indicates the solution on the convex hull which can be found by using a Lagrangian method.

The Lagrangian method changes the above budget constrained problem to an unconstrained problem by merging distortion and rate through a Lagrange multiplier λ, $\lambda \geq 0$. So (1.1) can be changed to

$$J_\lambda = \min_{q_i}(\sum_{i=1}^{N} (d_i(q_i) + \lambda \; r_i(q_i))), \tag{1.3}$$

where J_λ is the minimum cost under given λ. In the above equation, we can see that the minimum can be calculated for each data unit separately. Given λ, a point on the convex hull is such that the line of absolute slope λ is its tangent and will be the solution of (1.3) for that λ. Since the corresponding rate is monotonic

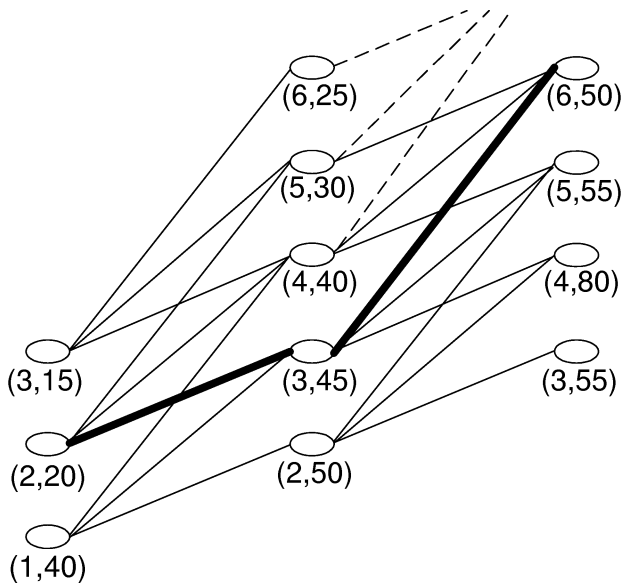
non-increasing with λ [64], the best solution can be found by solving the equation iteratively with smaller λ until the sum of rate is not over the total budget.

However, the operational rate-distortion curve is not convex in general and is not always non-increasing (see Fig. 1.4), where the operational rate-distortion curve means the rate-distortion boundary determined by using a given discrete set of quantization values and a coding scheme (see Fig. 1.5). Since the Lagrangian method finds only the solution on the convex hull, the optimal solution cannot be found if the operational rate-distortion curve is not convex and the solution is not on the convex hull. In Fig. 1.5, the best solution of the Lagrangian method is the point B under total budget R whereas the optimal solution of the original problem is the point A. While the Lagrangian method gives a suboptimal solution, this method is widely used due to its lower complexity. In video coding, this method is also used to find rate-distortion-optimized motion estimation vectors [71] and for rate-constrained motion estimation mode selection [80].

In order to find the global optimal solution, a dynamic programming method is used. Among the different types of dynamic programming methods, Viterbi algorithm [81] or Dijkstra's shortest path algorithm [7] can be used to solve this problem. Fig. 1.6 shows an example of a dynamic programming method. In the figure, a stage is each data unit, a state is the sum of bit-rates from the first stage to the current stage and a trellis is a connection from a state of the current stage to a reachable state of the next stage. The beauty of these algorithms

is that the optimal trellis path from a state (i) of the first stage to a state (j) of a current stage is part of global optimal trellis path from i to a state of the final stage through j. Therefore under the condition that the states (including cumulative costs and the path up to now) of a current stage are given, previous stages and future stages are independent. The complexity of this algorithm is $O(BNQ)$, where B is the total bit-budget, N is the number of data units and Q is the number of the quantization levels of each data unit. Comparing to the exhaustive search algorithm whose complexity is $O(Q^N)$, this method reduces complexity very significantly (although complexity remains very high). Sub-optimal methods such as those involving clustering of neighbor states [48] can be used to reduce complexity. In a clustering method, the number of states of each stage is reduced by merging several states into one state. Only the one path that has minimum distortion among those reaching the cluster is kept and the others are pruned.

Another application of a dynamic programming method in source coding is trellis-coded quantization (TCQ) [43] [12]. TCQ uses a structured codebook with an extended set of quantization levels and it reduces encoding complexity under a given level of performance. TCQ also gives a good result in wavelet image coding [67] and is adopted in JPEG2000 (Part II) [44].



r	d
1	30
2	20
3	15

r	d
1	25
2	20
3	10

r	d
1	40
2	30
3	5

Figure 1.6: Toy example of a dynamic programming method. Total bit-budget is 6, the number of data unit is 3 and each data unit has 3 quantization values. Thick lines indicate the optimal solution path and dashed lines indicate impossible paths due to limited budget. As we can see the optimal path between stage 1 to stage 2 which is found at stage 2, does not change after finding the global optimal path at stage 3.

1.2.4.2 Optimal solution under the MMAX criterion

When using the MMAX criterion and under the assumption that distortion is a non-increasing function of rate, finding the optimal solution is relatively simple because we can greedily choose a data unit that has maximum distortion and change its quantization index. This can be done until the total bit-budget is used up. For each data unit, since the states (operating points) are sorted in decreasing order of distortion, we need at most $N-1$ comparisons to find the maximum distortion in each iteration. There are in total NQ states so the algorithm is terminated in at most NQ iterations. Therefore the complexity is $O(N^2Q)$. A more efficient way to find the solution is by sorting all the states in decreasing order of distortion first and then doing iterations until the sum of rate is reached to the total bit-budget. Merge sorting can be used since states in each data unit are already sorted. The complexity of this sorting is $O(NQ \log N)$ (see Fig. 1.7.). The number of iterations is at most NQ and in each iteration, running time is $O(1)$. Therefore total complexity is $O(NQ \log N)$.

Another algorithm to find a MMAX solution is based on iteratively solving a minimum rate problem with a successively updated target distortion [59]. The following formula shows a minimum rate problem.

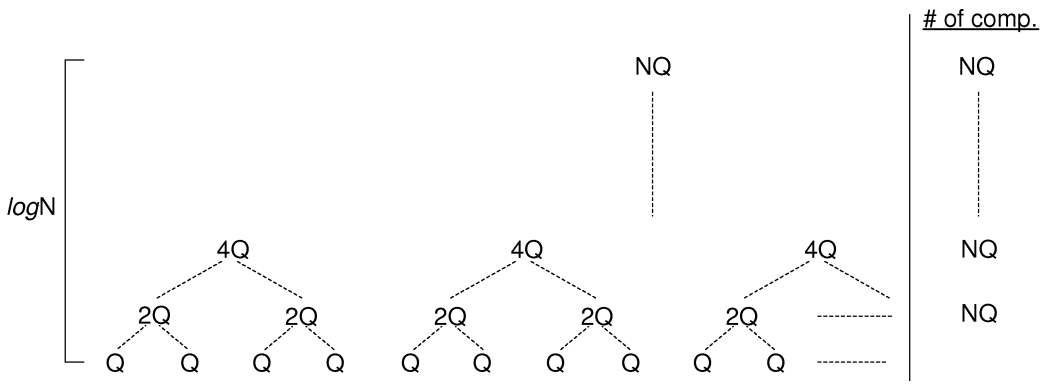


Figure 1.7: Iterative procedure of Merge sorting . $O(NQ)$ comparisons are needed in each level and there are $\log N$ levels. Therefore complexity of this merge sorting is $O(NQ \log N)$.

$$r_i = \begin{cases} \infty & : d_i(q_i) > D_{max} \\ r_i(q_i) & : d_i(q_i) \leq D_{max} \end{cases} \tag{1.4}$$

Using this equation, we find the sum of the rate of all data units whose distortion is smaller than or equal to the given D_{max}. If the sum of the rate is smaller than the total bit-budget then the target distortion D_{max} is decreased. Otherwise the target distortion is increased. This procedure is iterated until the total rate is the given bit-budget. A bisection method can also be applied but has the problem of potentially requiring infinite time to terminate, since the bisection is applied on the distortion which can be any positive real number. However this algorithm can be used to get a good initial point before applying a merge sorting algorithm.

After finding the MMAX optimal solution, an additional criterion such as MLEX or MMAX+ can be applied. In [58], these additional criteria are used to

break potential ties. In general, there can be two or more solutions that have the same rate sum and the same maximum distortion. Among these solutions, the one that has the best performance in an additional criterion is chosen as a final solution. In this thesis, the additional criterion is applied to use up the remaining bit-budget. (The toy example is given in the section 1.2.1.)

1.2.5 Bit allocation under a buffer constraint

In many video applications, encoding is done once by video content owners and decoding is done many times, each time a user decodes the sequence. Therefore, decoding has more physical or cost constraints than encoding. The memory size is one of these constraints and so an encoder should generate a bit-stream such that decoder buffer overflow is avoided while using a relatively small buffer. For this purpose, a Video Buffering Verifier (VBV) is used in MPEG1 [25] and MPEG2 [26] and, similarly, a Hypothetical Reference Decoder (HRD) is used in H.263 [29] and H.26L [54] [55].

This constraint can also be used in data transmisson through a constant bit-rate (CBR) channel under a delay constraint. In the case of CBR transmission with limited delay, the data in the encoder buffer cannot exceed $M \cdot C$ bits, where M is the maximum delay in seconds and C is channel rate in bits/second.

Otherwise, the data cannot be transmitted to meet the delay constraint of M seconds.

This problem is formulated as follows. Under the MMSE criterion,

$$J = \min_{q_i}(\sum_{i=1}^{N} d_i(q_i)) \text{ s.t. } B_i \leq B_{max}, \forall\ i = 1, \cdots, N \tag{1.5}$$

and under the MMAX criterion,

$$J = \min_{q_i}(\max_{i=1,\ldots,N}(d_i(q_i))) \text{ s.t. } B_i \leq B_{max}, \forall\ i = 1, \cdots, N, \tag{1.6}$$

where B_{max} is the buffer size and B_i is the buffer occupancy after coding the i^{th} data unit. B_i can be obtained recursively as

$$B_i = max(B_{i-1} + r_i - r_i^d,\ 0) \tag{1.7}$$

In (1.7), r_i^d is the amount of decoded data in the interval between the $(i-1)^{th}$ and i^{th} data units. The max operation is needed since the buffer state cannot be negative. r_i^d can be replaced by $C \cdot T$ in CBR transmission, where T is the interval between data units.

In [48], the optimal solution of this problem for the MMSE criterion is found by using a Viterbi algorithm. Also a fast approximation algorithm by using a Lagrangian method is proposed. For the MMAX criterion, the problem is solved

by the same method used in a budget constrained problem. The difference is that in this problem, for each iteration, B_i should be re-calculated to check whether any B_i violates the buffer constraint. Although the complexity of each iteration is increased, the complexity is still bounded as $O(NQ \log N)$ and this will be shown in chapter 4. The algorithm is terminated when any single violation is found, so in practice, the number of iterations is much smaller than that needed in a budget constrained problem.

The following formulation, where delay and channel constraints are considered, is a generalized version of this buffer constrained problem.

1.2.6 Bit allocation under delay and channel constraints

As explained in the previous section, the encoder buffer size is restricted by the maximum delay in CBR transmission. Therefore in this case, the objective of the bit allocation problem is finding the optimal solution under a given criterion without violating a buffer constraint. But in VBR transmission, several channel constraints are imposed in order to guarantee quality of service. For example, the constraints where a leaky bucket (or a token bucket) policing function is used are the size and output rate of the bucket. Peak transmission rate can be another channel constraint.

Optimal bit allocation algorithms for VBR video transmission over an ATM network are proposed in [5] [23]. In [5], leaky bucket policing function is used as a channel constraint and the Lagrangian method is used to find the optimal solution. Multiple buffer constraints are imposed in this case. An "anchor point" is defined, which indicates buffer overflow constraint when a single Lagrange multiplier is applied to find the optimal solution. After finding the anchor point, the Lagrangian method with a single Lagrange multiplier is used again to find the optimal solution up to this anchor point. This routine is iteratively applied until no more anchor points appears. Similar algorithms are used in [45] to find optimal bit allocation under multiple rate constraints. In [23], leaky bucket and double leaky bucket policing functions are considered as channel constraints and Viterbi algorithm is used to find the optimal solution. In the algorithm, the status of a leaky bucket and a decoder buffer is used as states in the algorithm. The violation of negotiated network transmission parameters and of the delay constraints can be checked by using the status of a leaky bucket and of a decoder buffer, respectively.

In [4], a MMAX criterion is used to achieve fair bandwidth allocation for VBR traffic in ATM networks. In this work, the MMAX criterion is not used at the encoder side to choose optimal bit-rate for each data unit. Instead it is used in the network queue side of wireless-ATM to decrease cell losses and delays.

Therefore a MMAX criterion is used for minimizing maximum buffer occupancy of each VBR source.

1.3 Contributions of this thesis

There are three issues related to image and video coding that are addressed in this thesis: capture, storage and transmission.

1. Chapter 2 studies a new image coding scheme that uses the special characteristics of the input devices such as digital cameras and digital camcorders. In a conventional scheme, the image processing and image compression stage are fully separated. So the objective of an image processing stage is maximizing visual quality of captured scenes and that of an image compression stage is minimizing source rates without severe distortion in visual quality. In this chapter, we consider a joint solution of image processing and image compression to reduce source rate with better visual quality. We focus on the Bayer color filter array, which is the most popular charge coupled device (CCD) in commercial input devices. The key idea to achieve our goal is to remove source redundancy that is added during image processing.

   Several methods such as format conversion, nonlinear transform and cropping are proposed to use popular image coding schemes like JPEG and

SPIHT. By using this algorithm, with same source rates, we can achieve better quality images in a whole range of compression ratios (when bi-linear interpolation is applied). Another important advantage is that this algorithm can reduce coding complexity by nearly 25% (50%) in color (monochrome) images and also reduce the blocking artifact in JPEG.

2. Chapter 3 focuses on the data storing problem in a situation where a fixed memory size is available. This chapter shows how to find an on-line bit allocation solution among the given number of images based on a minimization of maximum distortion criterion (MMAX). This problem is encountered in image input devices such as digital cameras. In digital cameras, each image has the same degree of importance, so MMAX is a proper criterion as an objective measure. To find an on-line solution, the statistics of future images need to be estimated by using training image sets and they are updated by using the statistics of previously captured images. Also we propose a T step delay method in which the statistics of the next T images are given before a current image is stored. A comparison with the solutions of constant distortion, constant quantization value and constant rate methods is provided to show the advantage of the proposed methods. From the comparison with the off-line optimal solution, we show that these methods give a near optimal solution.

3. Chapter 4 focuses on the optimal rate control for multimedia (especially, image and video) data transmission through an ATM network or a future IP network which provides quality of service. The problem here is how to maximize source quality without violating channel constraints. To find the optimal rate control scheme, a MMAX criterion and a MMAX+ criterion are used as a distortion measure. By using these criteria, minimum total distortion can be achieved under the condition that the minimum quality of each data is guaranteed. In this chapter, we provide the off-line optimal rate control of CBR and VBR transmission under token bucket policing based on MMAX and MMAX+ criteria. We also provide an algorithm to reduce the complexity of finding the MMAX+ solution.

Chapter 2

Image Compression in Digital Cameras with a Bayer Color Filter Array

In this chapter, we propose a new approach for image compression in digital cameras, where the goal is to achieve better quality at a given rate by using the characteristics of a Bayer color filter array. Most digital cameras produce color images by using one CCD plate and each pixel in an image has only one color component, so an interpolation method is needed to produce a full color image. After the image processing stage, in order to reduce the memory requirements of the camera, a lossless or lossy compression stage often follows. But in this scheme, before decreasing redundancy in a compression stage, redundancy is increased in an interpolation stage. In order to avoid increasing the redundancy before compression, we propose algorithms for image compression, in which the order of the compression and interpolation stages is reversed. We introduce image transform

algorithms, since uninterpolated images cannot be directly compressed with general image coders. The simulation results show that our algorithm outperforms conventional methods. This proposed algorithm provides not only better quality but also lower complexity because the amount of luminance data used in our method is only half of that in conventional methods.

2.1 Introduction

Digital cameras use image-processing schemes, e.g., interpolation techniques, such as those used in analog camcorders in order to achieve good quality images. One big difference between digital cameras and analog camcorders is that digital cameras store digital data in flash memories. Thanks to storing digital data, functionalities such as image editing and enhancement can be added. But the price of flash memories is still very high, so that the image bit-rates should be kept as low as possible (note that a 5 mega pixel CCD digital camera needs 15 Mbyte to store one image without compression). To do this, most digital cameras use lossy compression schemes like JPEG [49] to store the images. Therefore, like analog camcorders, most digital cameras improve image quality using image processing methods and then compress the image as much as possible while preserving perceptual quality. In other words, the image processing and image compression stages are completely decoupled.

In typical image processing stages redundancy is increased by color pixel interpolation. In order to produce full color images, most digital cameras place color filters on monochrome sensors. Some high-end digital cameras use three CCD plates to get full color images, where each plate takes one color component. But most digital cameras use one CCD plate with several different color filters and produce full color images by using an interpolation technique. Although there are several different color filter arrays (CFA) [82] [83], in this chapter, we focus on the Bayer CFA which is most widely used in digital cameras. The Bayer CFA shown in Fig. 2.1 uses 2 by 2 repeating patterns (RP) in which there are two green pixels, one red and one blue. There is only one color component in each pixel, so the other two color components for a given pixel have to be interpolated using neighboring pixel information. For example, in a bilinear interpolation method, the red (blue) color component on a green pixel in Fig. 2.1 is produced by the average value of two adjacent red (blue) pixels. Although there are several possible interpolation algorithms [50] [42] [75] [1] [33] [15], it is clear that from an information theoretic viewpoint they all result in an increase of redundancy.

In a conventional method, as shown in Fig. 2.2 (a), after finishing the image processing stage, a lossless or lossy image compression algorithm is used before storing the image. Although in theory one could achieve the same compression with or without the interpolation, to do so would require exploiting the

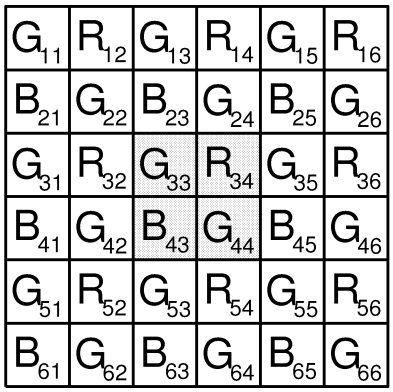


Figure 2.1: Bayer color filter array [3]. Each letter indicates the position of a different color filter. R, G and B are for Red, Green and Blue, respectively. The gray block indicates 2 by 2 repeating pattern.

specific characteristics of the interpolation technique with the compression algorithm. Clearly applying any standard compression method without modification does not exploit this knowledge. For this reason, in this chapter we propose image transformation algorithms to encode the image *before* interpolation, so that interpolation is performed only after decoding. In other words, the proposed algorithms compress images before adding the redundancy of the interpolation, as shown in Fig. 2.2 (b).

As an image coder, JPEG is widely used in digital cameras because it is relatively simple and provides good performance, especially when the compression ratio is low. But JPEG is a block discrete cosine transform (DCT) based coder and the blocking artifacts can become severe as the compression ratio becomes higher. Discrete wavelet transform (DWT) based coders such EZW [61], LZC [73], SPIHT [56], MTWC [78] and EBCOT [72] (adopted in JPEG2000 [74])

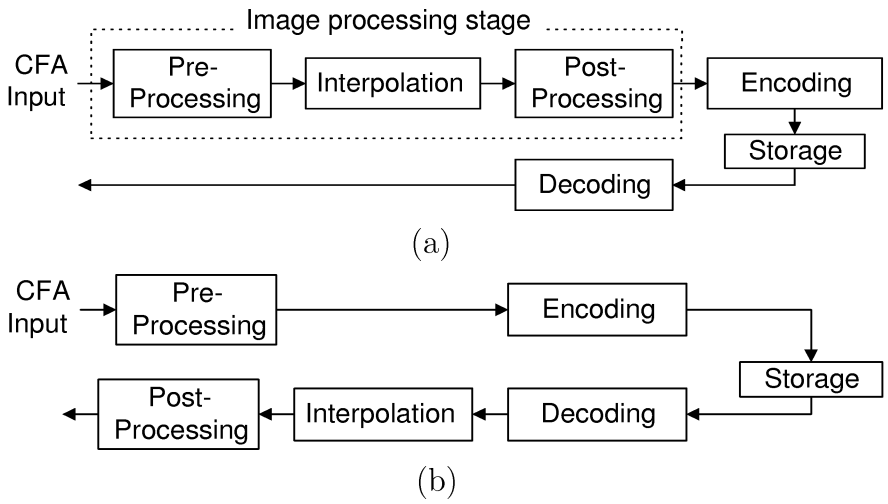


Figure 2.2: Block diagrams of (a) the conventional method and (b) the proposed method. In (a) an image processing stage is followed by a compression stage. In (b) interpolation and post-processing in an image processing stage are done after compression and decompression.

are also used as image coders. A DWT based coder does not produce blocking artifacts and it provides good performance under a high compression ratio. In this thesis, we use JPEG and SPIHT as representative of the DCT and DWT based approaches, respectively. Our proposed algorithms are tested under both of these coding techniques.

Methods to increase image quality using the redundancy of interpolation in post- and pre-processing stages have been studied. In [16], under the assumption of a fixed interpolation algorithm, the quantization noise is reduced by using an iterative method that incorporates information about the interpolation algorithm. By contrast our approach assumes only a specific CFA and can operate with any interpolation technique. Our approach is based on a novel technique to

map the existing image (with single color pixels) into an image that can be efficiently compressed. The main difference is that our algorithms compress non-interpolated images without introducing redundancy of interpolation, whereas the algorithm in [16] improves image quality by using the redundancy of interpolation.

In this chapter, extending our previous work in [35], we propose several different algorithms to transform the non-interpolated images before compression. We provide performance results of the proposed algorithms with different coders (JPEG and SPIHT) and interpolation methods (bilinear and adaptive interpolation). Also, using a simple example based on one-dimentional data, we provide an analysis to justify why our approach outperforms conventional methods.

This chapter is organized as follows: in section 2.2, the theoretical rate distortion performance of the conventional (compression after interpolation (CAI)) and proposed (interpolation after decoding (IAD)) approach is analyzed by using a 1-D sequence and DPCM encoding. Proposed image transformation algorithms are addressed in section 2.3. Experimental results are provided as demonstration of the validity of our algorithm in sections 2.4 and 2.5. Finally, the conclusion of this work is in section 2.6.

2.2 Performance comparison using one dimensional sources

Note that in our problem we do not have access to an original full color image, since the camera captures images with single color pixels. Thus, for the purpose of comparison we use as a reference a full color image obtained by interpolating the original (uncompressed) captured image. Therefore our problem will be to find coding schemes that are optimized in terms of minimizing the error with respect to that original interpolated image.

The main difference between the CAI and IAD methods is the order of compression and interpolation. In this section we show that an IAD method theoretically outperforms a CAI method by considering differential pulse code modulation (DPCM) compression of a one dimensional first order autoregressive (AR) process. Although open loop DPCM is not generally used due to error propagation in a decoded sequence, the difference sequence has an explicit theoretical R-D curve when the source is Gaussian AR processes. Therefore we consider open loop DPCM of one dimensional first order zero mean Gaussian AR processes. Let

$$X_n = \rho X_{n-1} + W_n, \ n = 1, 2, \cdots, \tag{2.1}$$

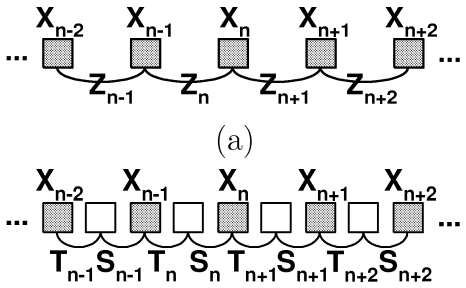


Figure 2.3: Gray and white boxes indicate original and interpolated samples, respectively. In (a), $\{Z_n\}$ is a differential sequence of the original sequence taken from sensors and in (b), $\{T_n\}$ and $\{S_n\}$ indicate a differential sequence of the interpolated sequence.

denote the process, where $\{W_n\}$ is a zero-mean sequence of independent and identically distributed random variables and $W_n \sim N(0, \sigma_W^2)$, and ρ is the correlaton coefficient ($0 \leq \rho < 1$). Then from the probability distribution of W_n, the probability distribution of X_n is $N(0, \sigma_W^2/(1-\rho^2))$. We assume that the initial state X_0 is given and we are interested in the source outputs for $n \geq 1$.

We define the differential sequence of $\{X_n\}$ as $\{Z_n\}$ then

$$Z_n \triangleq X_n - X_{n-1} = (\rho - 1)X_{n-1} + W_n \ . \tag{2.2}$$

Since X_{n-1} and W_n are independent, Z_n also has Gaussian distribution ($Z_n \sim N(0,\ 2\sigma_W^2/(1+\rho))$).

The rate distortion (R-D) function for a Gaussian source with mean square error (MSE) distortion can be written in closed form [8] and the R-D function of Z_n is

$$R_1(D) = \frac{1}{2}\log_2(\frac{2\sigma_W^2}{(1+\rho)D}),\ 0 \le D < \frac{2\sigma_W^2}{1+\rho}\ . \tag{2.3}$$

Next, we double the number of samples by using a linear interpolation method and define this new sequence as

$$\begin{cases} Y_{2n} \triangleq X_n, \\ \\ Y_{2n+1} \triangleq (X_n + X_{n+1})/2. \end{cases} \tag{2.4}$$

From this sequence, as shown in Fig. 2.3 (b), two differential sequences $\{T_n\}$ and $\{S_n\}$ are defined as

$$\begin{cases} T_n \triangleq Y_{2n-1} - Y_{2n-2} \\ \\ S_n \triangleq Y_{2n} - Y_{2n-1}. \end{cases} \tag{2.5}$$

Note that T_n is identical to S_n (i.e., $T_n = S_n = (X_n - X_{n-1})/2$) and the probability distribution of T_n (or S_n) is $N(0, \sigma_W^2/2(1+\rho))$. Since both $\{T_n\}$ and $\{S_n\}$ are Gaussian sources, the R-D functions are :

$$R_T(D) = R_S(D) = \frac{1}{2}\log_2(\frac{\sigma_W^2}{2(1+\rho)D}),\ 0 \le D < \frac{\sigma_W^2}{2(1+\rho)}\ . \tag{2.6}$$

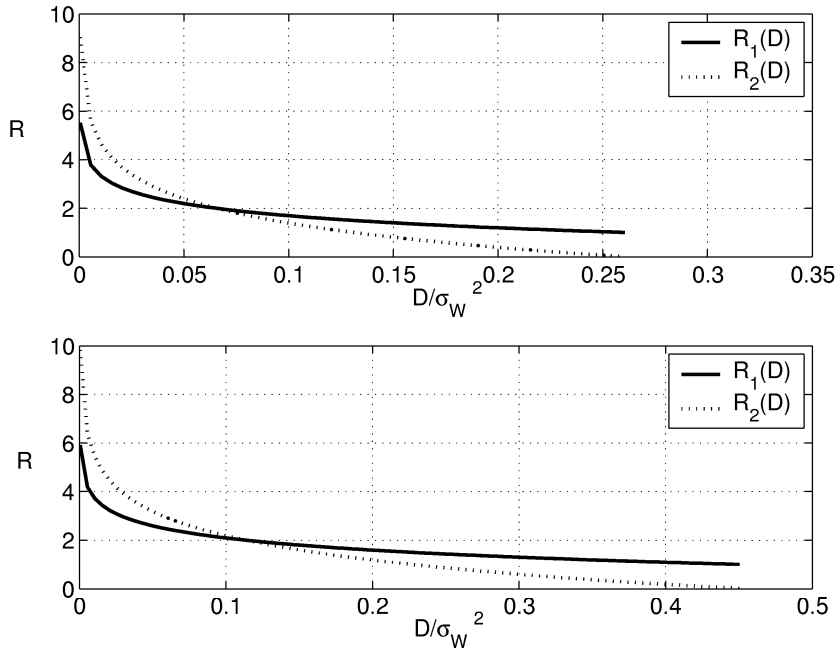


Figure 2.4: The upper (lower) graph is for $\rho = 0.9$ ($\rho = 0.1$). Solid lines indicate the R-D curve of the differential sequence ($\{Z_n\}$) and dotted lines indicate the R-D curve of the differential sequences after interpolation ($\{S_n\}$ and $\{T_n\}$).

In this example, since S_n and T_n are same, any information of S_n is not needed if T_n is provided. But in our original problem, the difference of neighboring pixels of interpolated images is not same and in the CAI method, the coder does not employ any additional information related to interpolation. Therefore we assume that the mutual information of S_i and T_j is not used during encoding for all i and j. Then the rate distortion function achieved with this method is :

$$R_2(D) = 2 \cdot R_T(D) = \log_2(\frac{\sigma_W^2}{2(1+\rho)D}),\ 0 \le D < \frac{\sigma_W^2}{2(1+\rho)}\ . \qquad (2.7)$$

The main difference between above two methods (DPCM and DPCM after in-

terpolation (DPCMI)) is the number of samples and the variance of sequences. The number of samples encoded by DPCM is half of the number encoded by DPCMI and DPCM has 4 times larger variance than DPCMI. The two methods have trade-off since the smaller number of samples provides better coding performance but the larger variance provides worse performance. Fig. 2.4 shows the performance comparison of two methods with different AR coefficients. In the figure, the performance of DPCM is better than that of DPCMI at higher rates but is worse at lower rates. Note that the rates corresponding to the intersection points of the two curves are the same but that the corresponding AR coefficients are different.

Next, instead of the theoretical rate distortion curves, we consider the R-D curves of general DPCM coders that use a uniform quantizer and an entropy coder. We assume that a given quantizer has N quantization bins. In the DPCM system, let each bin size be Δ (except top and bottom bins assuming that the range of source is infinite), let the average MSE be d and after entropy coding, let the average rate be r. In the DPCMI system, each bin size can be $\Delta/2$ and the average MSE is $d/4$ for T_n since the maximum sample value of T_n is half the maximum value for Z_n. This is because $T_n = (X_n - X_{n-1})/2 = Z_n/2$. However the number of samples in a given bin is exactly same as that in a corresponding bin of Z_n, so the rate is still r after applying the same entropy coder. Therefore, if the R-D curve of DPCM passes a point (r, d) then that of DPCMI passes a

point $(2r, d/4)$ (note that T_n and S_n each have rate r in the DPCMI system). This relation is formulated as

$$G(d) = \frac{1}{2}H(\frac{d}{4}), \tag{2.8}$$

where G and H are the R-D functions of DPCM and DPCMI, respectively. For instance, after encoding, if G is as in (2.3) then we can get (2.7) by using (2.8). As a result, the existence or location of the intersection point of DPCM and DPCMI depends on the function G. Note that the R-D performance of the difference sequences is different from that of source sequences since the decoder only has a quantized version of previous sample values. In an open loop DPCM case, the quantization error accumulates as the process continues, although theoretically the errors will cancel each other out in the long run [57]. But in transform coding which we use for image coding (i.e., DCT or DWT), there is no error accumulation and the distortion in a transform domain is the same as or close to that in a pixel domain depending on the transform. Therefore this performance comparison can be still valid for source sequences.

Note that the R-D function of the IAD method is calculated by using the source sequence instead of the reference sequence (i.e., interpolated sequence). The following shows that in the IAD method, the average MSE is decreased after interpolation. Let us assume that the average MSE and the mean of the

reconstructed sequence ($\{\hat{X}_n\}$) before interpolation are d and 0, respectively. Then the average MSE of interpolated samples is calculated as

$$E(\frac{\hat{X}_n + \hat{X}_{n+1}}{2} - \frac{X_n + X_{n+1}}{2})^2 = \frac{E(\hat{X}_n - X_n)^2 + E(\hat{X}_{n+1} - X_{n+1})^2}{4} = \frac{d}{2}. \quad (2.9)$$

Since the number of interpolated samples is identical to that of original samples, the average MSE after interpolation is reduced to $3d/4$. Therefore after interpolation, the cross point in Fig. 2.4 can be moved to right and this means that the proposed method provides better performance in a larger range of different compression ratios.

2.3 Image transformation algorithm to reduce redundancy

In order to avoid the redundancy introduced by CFA interpolation, we propose an algorithm in which an image compression stage precedes the CFA interpolation. The block diagrams of our IAD method and the CAI method are given in Fig. 2.2. There are some other functions, such as white balancing and color correction, that are performed in the image processing stage. These are shown as pre- and post-processings in the figure. From the figure, it seems like the two algorithms are almost same except the processing order, but the compression algorithm in each

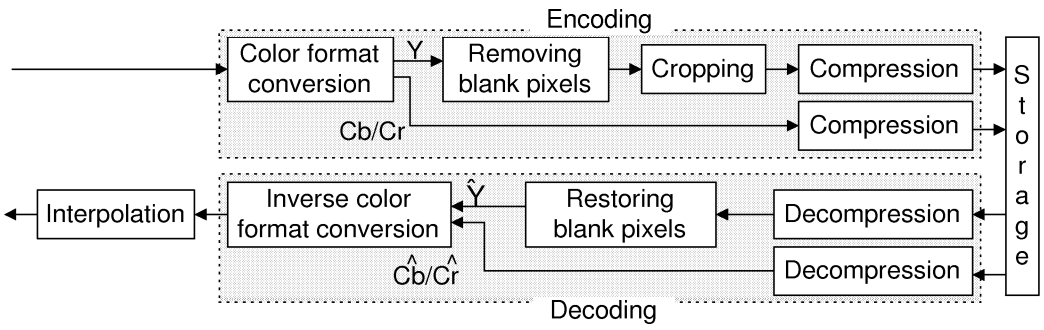


Figure 2.5: The detailed diagram of the encoding and decoding parts of the proposed method. Luminance (Y) data needs several transforms due to the location of them after format conversion, whereas chrominance (Cb/Cr) data can be coded directly.

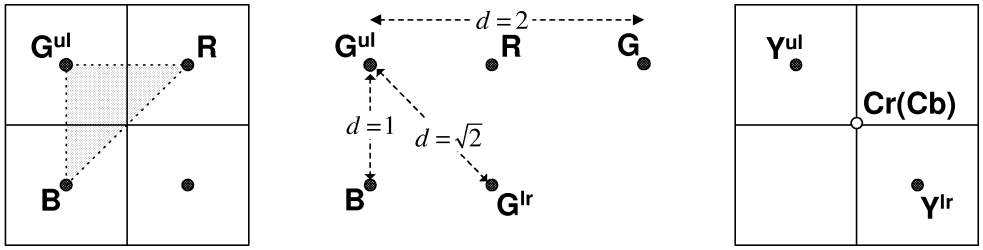
case is different because of the type of incoming data. Therefore our goal is to transform the input data into a format suitable for general image coders. The input data of our algorithm consists of only one color for each pixel, while in the CAI method there are three color values for each pixel, obtained by interpolation. In general image coders, it is assumed that incoming data is uniform (i.e., all pixels have the same color) and that the image has rectangular shape. Our goal is then to design a reversible image transform that can produce image data suitable for coding (without interpolation). A detailed version of the encoding and decoding blocks of the proposed method is shown in Fig. 2.5.

First, we propose a color format conversion algorithm since image coders usually use YCbCr format. After format conversion, luminance (Y) data is not available at every pixel, so that there are pixels that do not contain luminance information. In order to make the Y data compact, a transform which removes

those blank pixels is proposed. Then we show how to encode the resulting data which no longer has rectangular shape. In this chapter, JPEG and SPIHT are used in order to compare the performance of the CAI and IAD algorithms.

2.3.1 Color format conversion

In the CAI method, the data to be compressed has RGB format (obtained by interpolating the CFA data). This data is converted to YCbCr format before compression. In JPEG, normally 4:2:2 or 4:2:0 sampling is used and in JPEG2000, chrominant coefficients in high frequency bands after wavelet transform are not coded since human visual system is less sensitive to the chrominance data. In the proposed algorithm, to avoid increasing the redundancy, the number of pixels should not be increased after color format conversion. While there are several different methods to achieve this, we choose a method such that 2 green, 1 red and 1 blue pixels are converted to 2 Y, 1 Cb and 1 Cr pixel values. This is reasonable since luminance data is more important than chrominance data and the format conversion is reversible. We first propose a simple and fast method based on 2 by 2 blocks and then propose more complex methods that provide better performance.



(a) (b) (c)

Figure 2.6: The gray region in (a) indicates the possible location of Y data after the format conversion. (b) shows the distance between two green (or luminance) pixels. (c) shows the location of Y and Cr (Cb) data in a 2 by 2 block.

2.3.1.1 format conversion based on 2 by 2 blocks

In this format conversion, each 4 pixel block contains 2 green, 1 red and 1 blue pixels. Then two luminance and two crominance (i.e., Cb and Cr) are obtained as follows.

$$\begin{bmatrix} Y^{ul} \\ Y^{lr} \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & a_{13} & a_{14} \\ 0 & a_{11} & a_{13} & a_{14} \\ a_{31} & a_{31} & a_{33} & a_{34} \\ a_{41} & a_{41} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} G^{ul} \\ G^{lr} \\ B \\ R \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 128 \\ 128 \end{bmatrix}, \tag{2.10}$$

where, as shown inFig. 2.6 (b) and (c), $Y^{ul}(G^{ul})$ and $Y^{lr}(G^{lr})$ indicate luminance (green) data of the upper left corner and luminance (green) data of the lower right corner in 2 by 2 CFA block. The coefficients, a_{31} and a_{41} are half of standard coefficients of RGB to YCbCr conversion and the others are the same. The format

conversion matrix in (2.10) is invertible, by simply using the inverse matrix on the YCbCr data.

We now need to decide what the location of these YCbCr pixels should be. For the Cb and Cr data, each component could be located in any fixed position in the 2 by 2 block, since only one value of each chrominance is generated for the block. In the Y data case, however, one (Y^{ul}) should be located in the upper left region since Y^{ul} is the weighted average of G^{ul}, R and B (as shown in Fig. 2.6 (a)) and the other should be located in the lower right region of the block.

In our algorithm, we put the Y data at each green pixel position because green is roughly 60% of the Y data (the shape of the Y image is shown in Fig. 2.8 (a)). The location of the Y data is important, since improperly located Y data induces artificial high frequency components which can make the performance worse.

This method is simple and fast but YCbCr data of each 2 by 2 block depends only on the RGB data in that 2 by 2 block. Therefore, the YCbCr data potentially has more high frequency components than that generated by using bilinear interpolation (because each block is treated independently, while in the bilinear interpolation case each Y term is obtained from a larger set of pixels). The generated color components of bilinear interpolation are obtained by averaging the

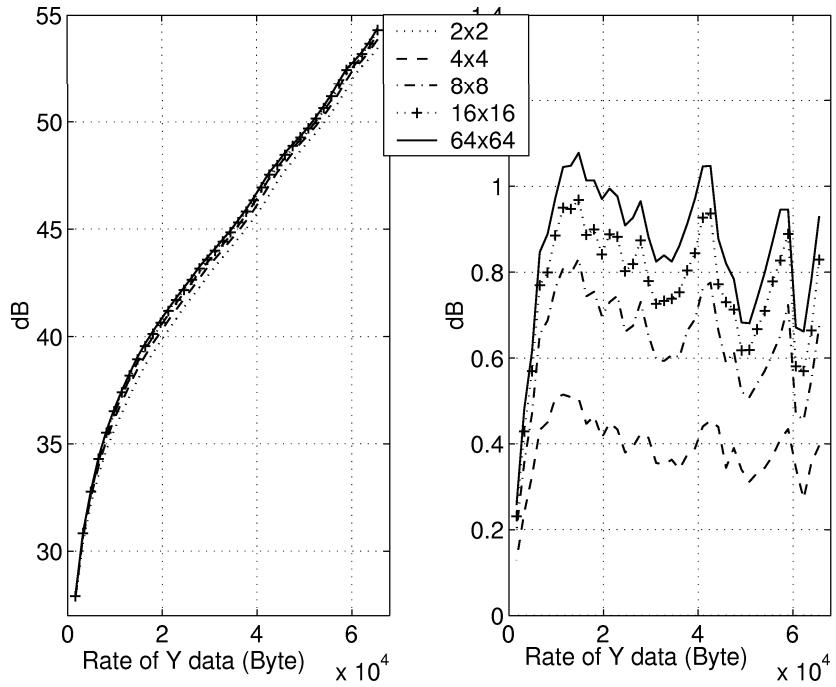
same color components located in neighboring pixels. For example, green and red color components at B_{43} in Fig. 2.1 are calculated as,

$$\begin{cases} G_{43} = \frac{(G_{42}+G_{44}+G_{33}+G_{53})}{4} \ , \\ R_{43} = \frac{(R_{32}+R_{34}+R_{52}+R_{54})}{4} \ . \end{cases} \tag{2.11}$$

2.3.1.2 Format conversion based on larger blocks

In order to generate smoother YCbCr data, we can consider a whole image as a block. After generating the RGB data for each pixel by using bilinear interpolation Y, Cb and Cr can be calculated from the RGB data on green, blue and red pixels respectively. These positions are chosen according to the degree of influence of each color (i.e., the dominant color components of Cb and Cr are blue and red, respectively). Because of the interpolation, the amount of RGB data is increased but the amount of YCbCr data is not increased, since each pixel position has only one component, either Y, Cb or Cr. This format conversion is also simple but the reverse format conversion is difficult due to the bilinear interpolation. To generate the original RGB data from the YCbCr data, a $w \cdot h$ by $w \cdot h$ reverse format conversion matrix is needed, where w and h are the width and height of an image respectively.

Although the decoding process (including reverse format conversion) can be done in a system with high computing power (e.g., personal computers), the



(a) (b)

Figure 2.7: (a) Coding performance comparison of Lenna image using different color format conversion methods. (b) Coding gain of the format conversion using larger blocks as compared to the format conversion with 2 by 2 blocks. Luminance data are coded by using SPIHT with shape adaptive DWT (SA-DWT) after rotation transform and the PSNR is calculated with Y and $\hat{Y}$ in Fig. 2.5 .

matrix is too large and the reverse conversion may still be too time consuming. In order to reduce the computational complexity, the above format conversion method can be applied to smaller blocks generated by dividing the source image. Since interpolation is done by using the pixels in the block, the column (or row) of reverse format conversion matrix is reduced to $W \cdot H$, where W and H are the width and height of a block respectively. The coding performance of the format conversion with different block sizes is shown in Fig. 2.7. Since the interpolated data at boundary pixels of each block is less smooth, the interpolation method

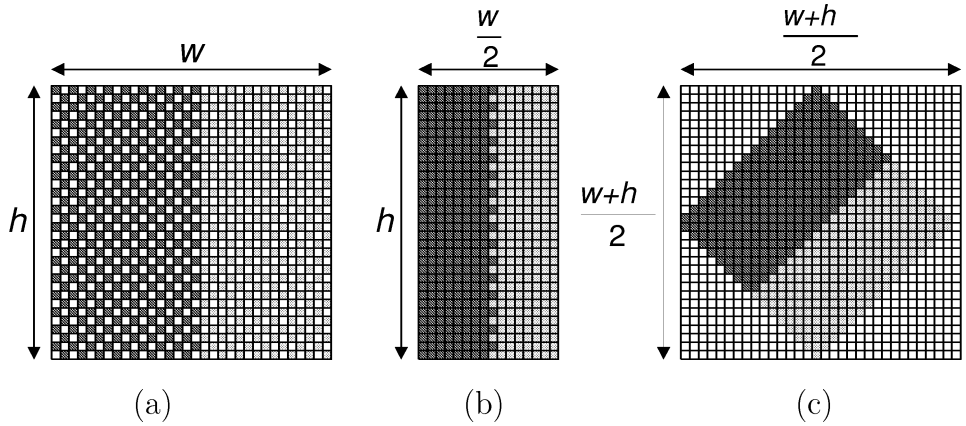


Figure 2.8: Transformation of Y (luminance) image. In the figure, dark and light gray pixels indicate Y data and white pixels indicate empty position. (a) indicates quincunx located Y image after format conversion, (b) and (c) indicate Y image after transform. In (b), each even column data is shifted to left odd column and in (c), each pixel is rotated 45 degree clockwise.

with a smaller number of boundary pixels can give a better result. For example, 75% of Y data are on block boundaries when 4 by 4 blocks are used whereas 12.3% are on block boundaries when 64 by 64 blocks are used. Therefore as shown in Fig. 2.7, the format conversion with larger blocks gives better results than that with smaller blocks although the complexity of decoding is higher.

2.3.2 Nonlinear transform to remove blank pixels

After the color format conversion, the Y values are not available all the original pixel positions (since the Y data is located only in the position of the green pixels), so general image compression methods cannot be directly applied to compress the Y image. Therefore another reversible transform is needed to change the

quincunx located Y pixels to normally located Y pixels (i.e., so that we obtain a Y image with no blank pixels). As in Fig. 2.8 (b), one possible simple transform is a horizontal pixel shift where pixels in even columns are shifted to the left odd column and all even columns are removed. This transform can be formulated as

$$if\ x+y=odd,\ \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{cases} \begin{bmatrix} \frac{x}{2} \\ y \end{bmatrix}, & if\ x=even, \\ \begin{bmatrix} \frac{x-1}{2} \\ y \end{bmatrix}, & if\ x=odd, \end{cases} \tag{2.12}$$

where (x, y) and (X, Y) are the pixel positions in the images before and after transformation, respectively. Here we assume that the origin is the lower left corner of an image. A vertical shift transform can be similarly defined, but we focus here on the horizontal shift transform.

After the transform is performed, a vertical edge causes artificial high frequency components in vertical direction and this makes coding performance worse. Note that if a vertical shift had been chosen the same problem would arise with respect to horizontal edges. Thus, under JPEG coding with a high compression ratio, most of this artificial high frequency information may be lost. In Fig. 2.8 (b), the line shaped boundary between dark and light gray is changed to a zigzag shaped boundary. Spatially weak correlation is another reason to

make the result worse. If the distance of adjacent pixels in a CFA is assumed to be 1 then, after horizontal shifting, the vertical and horizontal distances of adjacent pixels in the Y data are $\sqrt{2}$ and 2 respectively (see Fig. 2.6 (b)).

An alternative simple transform to remove blank pixels among Y data, which does not pose these problems, is 45 degree rotation formulated as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{1}{2}\left(\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -1 \\ w-1 \end{bmatrix} \right), \quad if\ x+y = odd, \tag{2.13}$$

where w indicates the width of an image. As shown in Fig. 2.8 (c), after rotation, Y data is concentrated on the center of an image with a oblique rectangular shape. This transform does not induce artificial high frequencies and the vertical and horizontal distances of adjacent pixels are now $\sqrt{2}$. But since the data are in an oblique rectangular shape area, some redundancy is added when the boundary pixels are coded. This is addressed in the next section.

The performance of the two methods after coding is shown in Fig. 2.9. Since, in JPEG coding (shown in (a)), high frequency components introduce more errors due to large quantization values, the result of the method inducing more high frequency components is worse. Also in JPEG coding, an image is efficiently coded by using EOB (end-of-block). When an image has additional high frequency components, the EOB occurs later in the zigzag scan on average and

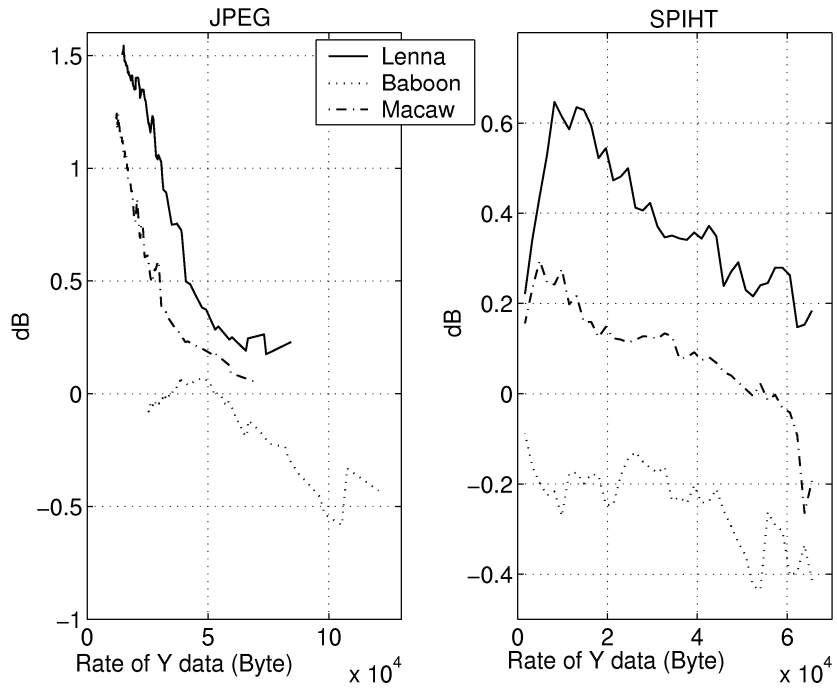


Figure 2.9: PSNR difference of luminance data between the rotation and horizontal shift methods after compression by using (a) JPEG and (b) SPIHT. 2 by 2 block format conversion is used in both cases.

thus the overall rate tends to increase. As expected, the horizontal shift transform generates more high frequency components and gives a worse result. For the "Baboon" image, however the image itself contains significant high frequency information and most frequency coefficients have to be coded (i.e., EOB coding does not help much), therefore the induced high frequency components by the shift method result in a less significant panalty than for other images. Also, due to the added redundancy coming from the data shape of the rotation method, the result of the shift method can be better in cases where the artificial high frequency components do not have a larger effect on the coding result (as in the

"Baboon" image). Contrary to JPEG coding, SPIHT uses the same quantization for all frequency bands but achieves compression by transmitting bit-planes in order of significance. Therefore the coding performance of the shift method is only slightly worse than that of the rotation method. But if the source is simple (i.e., not having large energy in high frequency bands) then the shift method provides worse energy compaction and so the result is worse (as shown in the case of "Lenna" image).

In Figs. 2.9 (a) and (b), the coding gain of the rotation method is decreased as the bit-rate is increased, except at very low bit-rate. In the low bit-rate region, small coefficients are quantized to zero, therefore most of coefficients are not transmitted because an EOB has been reached (in JPEG coding) or only a small number of coefficients is coded (in SPIHT coding). But in the shift method, many coefficients are large and cannot be quantized to zero. Therefore higher coding gain is achieved with the rotation method. As quantization values become smaller, the coefficients of the rotation method (which are quantized to zero in a low bit-rate region) are no longer quantized to zero and the bit-rate increases sharply. Instead, most coefficients of the shift method are already non-zero (in the low bit-rate region) and the bit-rate is increased more slowly. Therefore the coding gain of the rotation method is reduced as the bit-rate becomes higher. Although in the "Baboon" case, the performance of the shift method is better, this is due to the redundancy of the rotation method, so the coding gain is

decreased more as bit-rate becomes higher. In SPIHT coding with very low bit-rate, only a few most significant bit-plane can be coded and the coded data are similar to each other, therefore no coding gain is achieved.

2.3.3 Data cropping for images obtained by the rotation transformation

After the horizontal shift transformation, the shape of Y data is still suitable for coding as in Fig. 2.8 (b). But as in Fig. 2.8 (c), the shape of Y data after the rotation transform is not rectangular and thus coding the whole rectangular region that includes the oblique rectangular shape of Y data would result in some inefficiency in the coding. Therefore a proper cropping method is needed to remove the data outside of the oblique rectangular area containing Y data.

2.3.3.1 Data cropping for JPEG (DCT based coders)

In JPEG, the size of a DCT block is 8 by 8 and blocks that consist of blank pixels only (blank blocks) do not need to be coded. In addition, we do not need to send any side information about the location of Y data since it can be calculated at the decoder by using the size of images. As shown in Fig. 2.8 (c), the number of blank blocks depends on the width and height of the image and 6 bits (2 bits for a zero DC value and 4 bits for EOB) are needed to code a blank block when

standard Huffman tables of JPEG are employed. In case of 512 by 512 images, out of 4096 blocks, the number of blank blocks is 1984 and without coding blank blocks, we can save 1488 bytes. The blocks containing boundary pixels of Y data (boundary blocks) also contain blank pixels since Y data are in an oblique rectangular shape. As a result, compared to the shift method, the number of blocks to be coded is increased by $(w + h)/16$ in case that the width and height are multiples of 16.

Proper padding methods are needed for boundary blocks since the discontinuity between blank and data pixels in the block creates artificial edges that require a significant coding rate. Because boundary blocks have Y data only in the position of an upper or lower triangular region, padding can be simply done by diagonal mirroring. Better performance can be achieved by using low-pass extrapolation (LPE) [32] or shape adaptive DCT (SA-DCT) [65][32][70]. LPE is relatively simple and provides good R-D performance whereas SA-DCT provides better performance but is more complex. In our case, after the rotation transform, The data pixels in the boundary blocks are always in a triangular region of the block corner. Since data pixels can be moved to the upper left corner by simple rotation, SA-DCT can be easily applied. Also after SA-DCT, only DCT coefficients in the upper left triangular region are non zero so that an EOB can be inserted early on the zigzag scan.

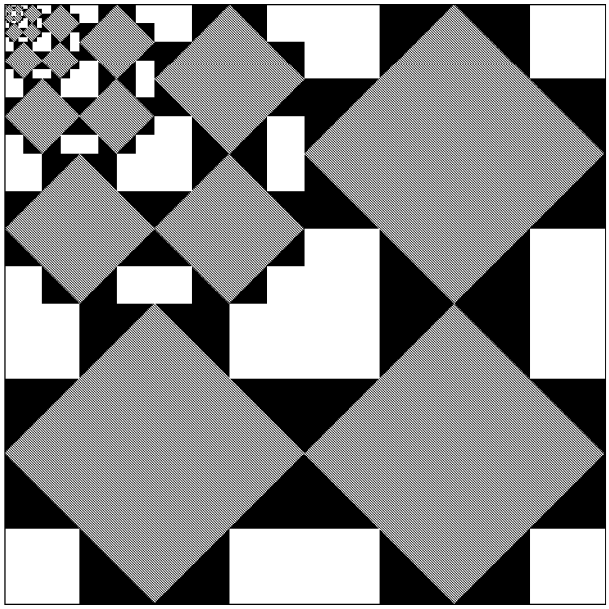


Figure 2.10: The coefficients map after SA-DWT. Gray regions indicate meaningful coefficients and black and white regions indicate blank coefficients.

2.3.3.2 Data cropping for SPIHT (DWT based coders)

Contrary to JPEG, SPIHT is not a block based coder and so the method used with JPEG cannot be applied. Therefore we need to introduce new coding method in order to code Y data in the oblique rectangular area only. In the still image coding of MPEG-4, arbitrarily shaped objects are coded by using shape adaptive DWT (SA-DWT) [38]. One of the good features of SA-DWT is that the number of coefficients after SA-DWT is identical to the number of data pixels. In order to code data pixels only, we employ SPIHT with SA-DWT. But without modifying entropy coding in SPIHT, some redundancy is still added since SPIHT uses a two by two block arithmetic coding algorithm.

In Fig. 2.10, only the gray regions contain meaningful coefficients after SA-DWT. Out of 16 two by two blocks in the lowest frequency band, only 4 blocks located in each corner consist of blank coefficients. Since all descendants of these blocks (white regions in the figure) are blank coefficients, these regions are not coded. But blank coefficients in black regions in the figure are involved in coding due to the entropy coding scheme of SPIHT and since they are not skipped some redundancy is introduced.

As a consequence of the redundancy induced by coding blank data in JPEG and SPIHT, the shift transform outperforms the rotation transform in some cases as shown in Fig. 2.9.

2.3.4 Influence of chrominance data over luminance data

Fig. 2.11 shows that the IAD algorithms also give a better result in chrominance data coding. In the IAD algorithms, one Cb (Cr) data is chosen out of 4 CFA pixels and the width and height of the Cb (Cr) image are $w/2$ and $h/2$, respectively. Therefore the data size is reduced to a quarter of that of the conventional technique whereas the pixel distance is doubled.

But contrary to the CAI method, in which the coding results of luminance and chrominance data are fully separated, chrominance data with large distortion can add distortion to luminance data after interpolation and vice versa.

For the case of format conversion with 2 by 2 blocks, the coding error in RGB data is calculated as follows.

$$\begin{bmatrix} e(G^{ul}) \\ e(G^{lr}) \\ e(B) \\ e(R) \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & a_{13} & a_{14} \\ 0 & a_{11} & a_{13} & a_{14} \\ a_{31} & a_{31} & a_{33} & a_{34} \\ a_{41} & a_{41} & a_{43} & a_{44} \end{bmatrix}^{-1} \begin{bmatrix} e(Y^{ul}) \\ e(Y^{lr}) \\ e(Cb) \\ e(Cr) \end{bmatrix}, \tag{2.14}$$

where $e(\cdot)$ is the error of each component due to lossy coding. Since the final Y data (after interpolation) is calculated from the distorted RGB data, i.e., from distorted YCbCr data, the error in the final Y data depends on the quantization errors in both Y and Cb (Cr) data.

Fig. 2.12 shows that the influence of chrominance error is larger in the case of 2 by 2 format conversion. The reason is that the Cb (Cr) data after the 2 by 2 format conversion has more high frequency components which means that at the same rate the distortion is higher than if a 64 by 64 block is used. Also in this figure, the PSNR of chrominance data with high bit-rate Y data drops more than that with low bit-rate Y data since in the case of high bit-rate Y data (the upper curve), most of errors are induced from the error of chrominance data.

2.3.4.1 Bit allocation between luminance and chrominance data

In SPIHT, each component is coded separately and there are no explicit mechanisms for bit allocation, whereas in JPEG bit allocation to each component cannot be explicitly controlled and is determined by the chosen quantization tables and the data characteristics. Therefore in SPIHT, it is necessary to determine the bit allocation between luminance and chrominance data based on human visual sensitivity to each component. Moreover, in the proposed methods, the bit-rate of one component affects the quality of the other components, so the overall performance is changed depending on the bit allocation.

Here, we simply consider bit allocation based on the quality of luminance data since the human visual system is more sensitive to luminance data. Fig. 2.13 (a) shows the quality change of luminance data after interpolation depending on the overall bit-rate and the bit-rate of the luminance data. In the figure, the lower curves are fairly flat, but the upper curves drops sharply and some curves intersect. This means that the effect of chrominance data is larger when the bit-rate of luminance data is higher. Thus under a certain bit budget constraint, we need to decrease the bit-rate of luminance data to maximize the quality of luminance data. As in Fig. 2.13 (a), if the Y data can be coded with only 6 different rates, then, from the figure, we can choose the best bit allocation. This means that if we have the results of all possible rates, we can find the optimal

bit allocation under any given bit budget. Similar to Fig. 2.13 (a), (b) shows the quality change of chrominance data after interpolation. Although the curves drop sharply, this happens in the range where the bit-rate of luminance data is relatively low comparing with that of chrominance data. In general, the bit-rate of luminance data is higher and this drop does not have big effect under proper bit allocation (i.e., the bit-rate of luminance data is higher than that of chrominace data). This also guarantees that we can focus on the quality of luminance data since under the proper bit allocation, the quality change of chrominance data is small.

Another conclusion we can make from the figure is we do not need to be too concerned with the bit allocation when overall bit-rate is not very high (as shown in Fig. 2.13 (a), only the curves corresponding to high bit-rate intersect). Since the R-D characteristics are different for each image, we fixed the bit-rate of Cb (Cr) data to be a quarter of that of Y data.

2.4 Experimental results and comparison

In order to confirm the validity of the IAD algorithms (i.e, horizontal shift transform with 2 by 2 block format conversion and rotation transform with 2 by 2 and 64 by 64 block format conversion), we implemented these algorithms and compared the results with those obtained with CAI (JPEG with 4:2:2 format

and SPIHT) methods. Due to the lack of CFA raw data, we generate CFA raw data by using test images such as "baboon", "lenna" and "macaw"(H : 512, W : 512, 24 bit color, 786.432KB). Actually, what we obtain is not CFA raw data, since in these data all image processing functions, except interpolation, have been already done. But, in this chapter, we mainly focus on interpolation and compression methods, so other image processing parts are not considered. Our results are the same as the results achieved when all image processing functions are done before compression and interpolation. The results of the proposed and conventional algorithms are compared using the PSNR of luminance and chrominance data at each target bit-rate. As we mentioned in section 2.2, in order to calculate PSNR, we consider bi-linear interpolated images without compression as the source images. The CAI method compresses this interpolated image and the IAD methods use the same interpolation after decompression.

As shown in Fig. 2.14 (a),(c) and (e), the IAD algorithms achieve better luminance PSNR under all different bit-rates except for very low bit-rates. With JPEG compression, the PSNR of the shift method drops sharply and the performance of this method is worse than that of CAI methods in case that the bit-rate is approximately under 50KB (i.e., the compression ratio is roughly 15 : 1) whereas the rotation methods outperform the CAI method under all other compression ratios used in [14]. With SPIHT compression, the performance of shift and rotation with 2 by 2 block format conversion methods is similar (see Fig. 2.9)

and they outperform the CAI method when the bit-rate is over 20KB or 25KB (i.e., a compression ratio is 39 : 1 or 31 : 1) as shown in Fig. 2.15. As expected, the rotation with 64 by 64 block transform method gives a better result than other proposed methods but it needs high computing power at the decoder side. Under same bit-rate, IAD algorithms can use a lower compression ratio (or higher bit-rate per pixel) since the IAD algorithms only use approximately half of luminance data. This is the reason why the IAD methods outperform the CAI method.

As shown in Fig. 2.14 (b), (d) and (f), in chrominance data cases, the PSNR gain is even higher (though PSNR is not so meaningful in color components.). In the CAI algorithm, if the 4:2:2 format is used for JPEG compression then two adjacent pixels use same color data and the some color information is lost. But in the IAD algorithm, color format conversion is reversible and all color information can be presented. Although, even in the CAI method, there is no color information loss if JPEG with 4:4:4 format is used, the bit-rate for the color information is increased and so, by using this increased bit-rate, lower compression ratio can be applied in the IAD algorithm. In our experiments, chrominance data compression with 4:4:4 format (i.e, no data loss during format conversion in the CAI algorithm) is tested with SPIHT compression. Since the size of chrominance data of the CAI algorithm is 4 times larger than that of proposed ones, the bit

budget per pixel of IAD algorithms is 4 times larger than that of conventional one and this gives a large PSNR gain.

Although shift and rotation with 2 by 2 block transform provide exactly same chrominance data (since both transforms use the same 2 by 2 color format conversion), the chrominance PSNR of both algorithms are not identical. This shows that the luminance distortion also affects the quality of chrominance data.

From Fig. 2.14, one common phenomenon is that the PSNR gain becomes larger as the bit-rate increases. This can be explained by the color format conversion and spatial correlation. In the IAD algorithms with 2 by 2 block format conversion, the Y data in green position is generated by using only one green at the position and one red and blue neighboring pixels to recover each color component after decompression. This conversion creates more high frequency components than that of a conventional bilinear interpolation method.

Also, more high frequency components are introduced by lower spatial correlation. DCT blocks of Y data in our algorithm correspond to a larger area and therefore have weaker spatial correlation (under the assumption that an image has strong spatial correlation.) [31]. As a result, the proposed algorithm has more high frequency components which leads to higher distortion under JPEG compression, since in JPEG high frequency components are quantized more coarsely. As the compression ratio becomes lower, the CAI method can keep more high frequency components and so the bit-rate of the compressed image increases rapidly.

Highly spatially correlated images have small high frequency coefficients which become zero under high compression ratio and leads to efficient coding (with EOB). But if the compression ratio becomes lower, then high frequency coefficients components have non-zero values after quantization (even if they are small) so they cannot be efficiently compressed. Therefore we can get higher PSNR gain (the difference between IAD algorithms and a CAI algorithm) as compression ratio becomes lower. This can also explain that the "Baboon" image (low spatial correlation) has higher PSNR gain than the "Lenna" and "Macaw" images (high spatial correlation) under high compression ratio.

In SPIHT, DWT coefficients are coded from the highest bit-plane and so large high frequency coefficients can be coded without resulting in large increases in distortion even at low bit-rate coding. But due to weak correlation, the IAD algorithms generate larger high frequency coefficients and cannot be efficiently coded in low bit-rate coding. Similar to the result of JPEG coding, the "Baboon" image has a higher PSNR gain than other images since the image has weaker correlation than other images.

In addition to the higher PSNR gain and lower complexity, the IAD algorithms have other advantages such as lower blocking artifact after JPEG coding and fast consecutive capturing. Lower blocking effect under the same bit-rate is achieved by lower compression ratio, interpolation and different block shapes. Because interpolation is done after decompression, this function can reduce blocking artifact

similar to a de-blocking processing after JPEG decompression. Luminance data and chrominance data use different shapes, which may also help to reduce blocking artifacts. Also fast consecutive capturing is possible since the compression time is shorter by compressing only around half of Y data and interpolation (in case of 2 by 2 format conversion) and post processing functions are not needed during the capture process.

2.5 Comparison with adaptive interpolation

From the compression viewpoint, the bilinear interpolation is a good method because it results in smoother (and thus easier to compress) images. Also in the IAD algorithms, the color components generated by using the bilinear interpolation have an error that results from averaging the error of neighbor pixels, so that the average distortion of generated color components is lower than that of coded color components (similar to the 1-D case shown in (2.9)). This is also confirmed by the experimental results shown in Fig. 2.7 (a) and Fig. 2.14 (c) (SPIHT). Note that the two figures have different horizontal axis and the rate used in Fig. 2.14 is 1.5 times larger than that of Fig. 2.7. The result verifies that PSNR is increased after interpolation except at high bit-rates (where round-off error plays an important role due to small coding error).

Although the bilinear interpolation is simple and fast, it works like a low pass filtering and does produce smoothing of edges. To preserve more edge information, several different adaptive interpolation algorithms have been proposed. Depending on the local information, adaptive interpolation algorithms take a different interpolation method and use the correlation of different color components. After applying the adaptive interpolation, the interpolated image has more edge information (i.e., more high frequency components) and it cannot be easily compressed. In this sense the IAD algorithms have an advantage. Note that the IAD algorithms perform interpolation after decoding, so the coded data is independent of interpolation algorithms. But due to lossy compression, IAD and CAI algorithms have different data before the interpolation. Therefore it could happen that they have different edge information and take a different directional interpolation method for pixels at the same position. This makes generated color components have large distortion. Also, the error of one color component is involved in the interpolation of other color components and the distortion of generated pixels can be increased. As a result, by using the adaptive interpolation, the IAD algorithms achieve some gains from data smoothness before compression (especially in the case of rotation with 64 by 64 block format conversion) but may lose in perpermance from choosing different directions during interpolation due to distorted data.

To verify the performance of the IAD algorithms with the adaptive interpolation, we consider 3 different adaptive interpolation algorithms constant hue-based, gradient based and median-based interpolation [50].

Constant hue-based interpolation is proposed by Cok [6] and Kimmel [33], where hue is defined by a vector of ratios as $(R/G,\ B/G)$. In this algorithm, the green color component is used as a denominator and a small error of the green component may induce a large error in hue, especially when green values are small. Therefore the IAD algorithms do not provide good performance when this interpolation is applied.

Gradient based interpolation is proposed by Laroche and Prescott [34]. In this algorithm, at first, green components on blue (red) pixel positions are determined by using the directional bilinear interpolation, where the direction is selected by the gradient of neighboring blue (red) components. After determining green components, blue (red) components are interpolated from the differences between blue (red) and green components. Fig. 2.16 shows the coding results of IAD algorithms. With JPEG compression, the performance of IAD algorithms (except rotation with 64 by 64 block format conversion) is worse than that of the CAI algorithm since different direction is determined by large error in high frequency components and blocking effect and the error of green components also affects red and blue components. But with SPIHT compression, the coding error is evenly distributed and different directional interpolation is reduced. Therefore as shown

in Fig. 2.16 (d), the IAD algorithms outperform the CAI algorithm although the gain is smaller than when bilinear interpolation is used (shown in Fig. 2.15).

The performance is also tested with Median-based interpolation (proposed by Freeman [13]) which employs two step processes. The first pass is the bilinear interpolation and the second pass is selecting the median of color differences of neighboring pixels. Fig. 2.17 shows the coding results of IAD algorithms with 3 by 3 median filter. Similar to the gradient-based interpolation, the IAD algorithms provide worse results when JPEG is applied. But with SPIHT, IAD algorithms still provide better results up to $20:1$ or $40:1$ compression ratio depending on the format conversion methods.

As a result, the IAD algorithms with SPIHT provide better results with the gradient based and median-based interpolation. But due to coding inefficiency, irregular coding error and blocking effect, the IAD algorithms with JPEG take the different direction of interpolation for each pixel, therefore the performance is worse.

2.6 Conclusion

In this chapter, we investigated the redundancy decreasing method by merging an image processing stage and an image compression stage. Several color format

conversion algorithms and shift and rotation transforms are introduced to compress CFA images before making full color images by interpolation. We showed that proposed algorithms outperforms that of the conventional method in full range of compression ratio of JPEG coding with the bilinear interpolation and up to 20 : 1 or 40 : 1 compression ratio (depending on the color format conversion and interpolation methods) with SPIHT coding when the bilinear, gradient based and median-based interpolation are applied. Also we analyzed the reason that PSNR gain becomes higher as compression ratio becomes lower and checked it with a 1D DPCM sequence. Because the proposed algorithms use only around a half size of Y data and only need additional simple transform, the computational complexity can be decreased. Also, with this algorithm, reducing blocking artifact and fast consecutive capturing can be achieved.

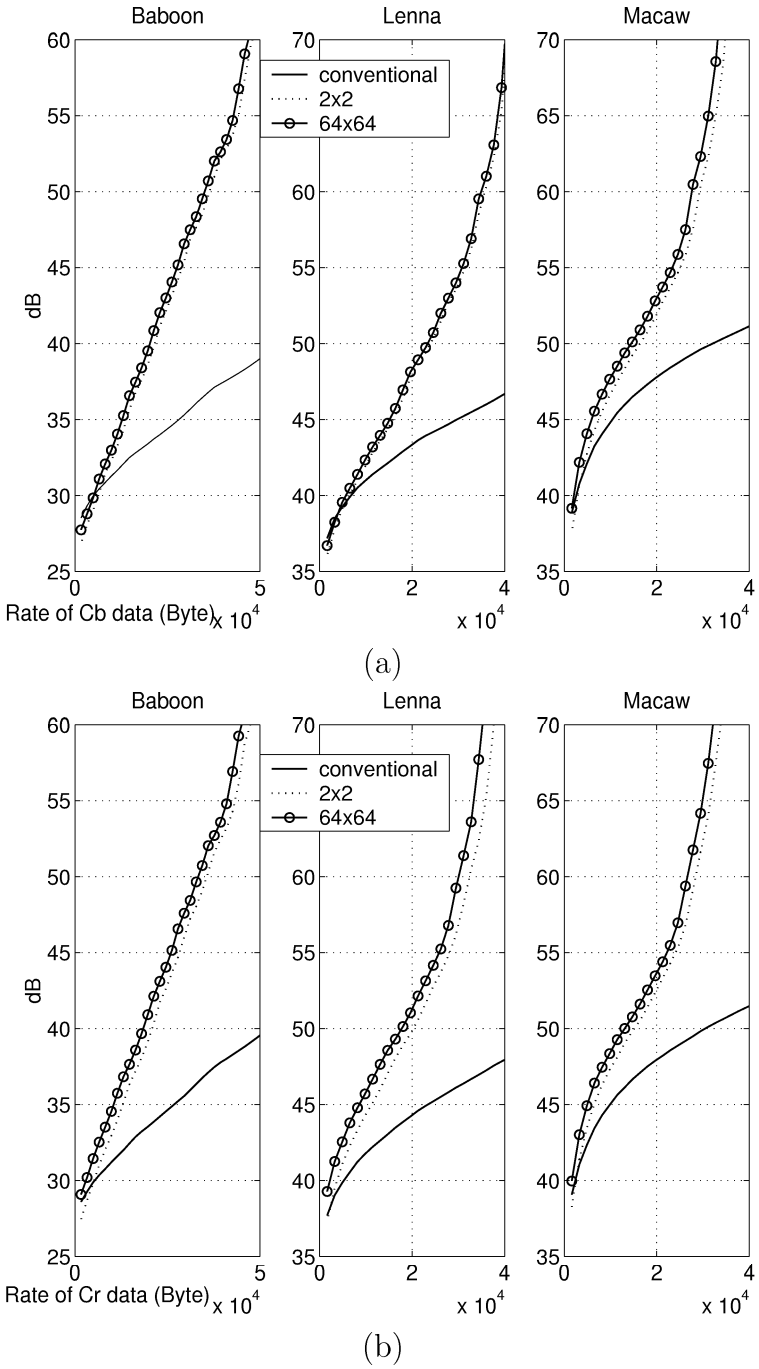


Figure 2.11: Coding performance of chrominance data ((a) Cb and (b) Cr) of CAI and IAD algorithms. SPIHT is used as a compression method. R-D data are calculated from $\hat{Cb}$ and Cb ($\hat{Cr}$ and Cr) in Fig. 2.5.

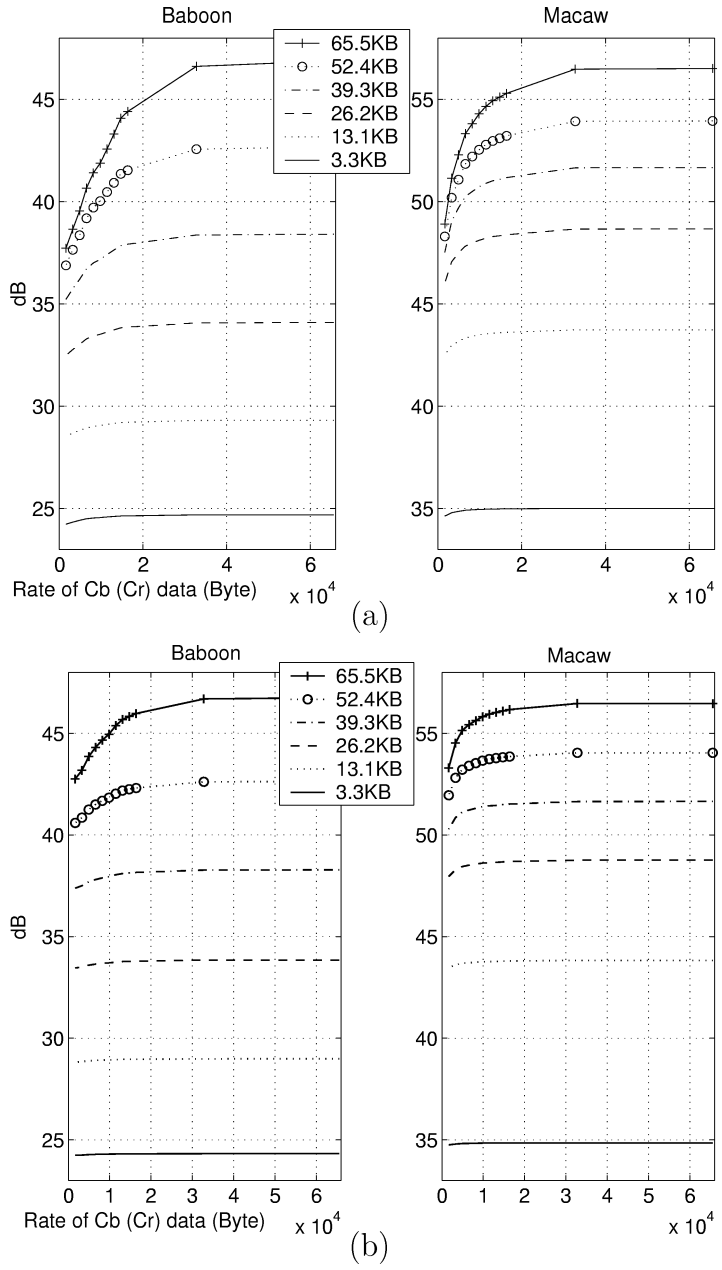


Figure 2.12: Effects of the distortion in chrominance data on the distortion in luminance data after interpolation. 2 by 2 block and 64 by 64 block format conversion is used in (a) and (b) respectively. Each curve corresponds to Y data coded with different bit-rate. The vertical axis indicates the PSNR of Y data and the horizontal axis indicates the rate of Cb (Cr) data. SPIHT is used as a compression method and PSNR is calculated from the distortion between the interpolated image before compression and the final output image of proposed methods.

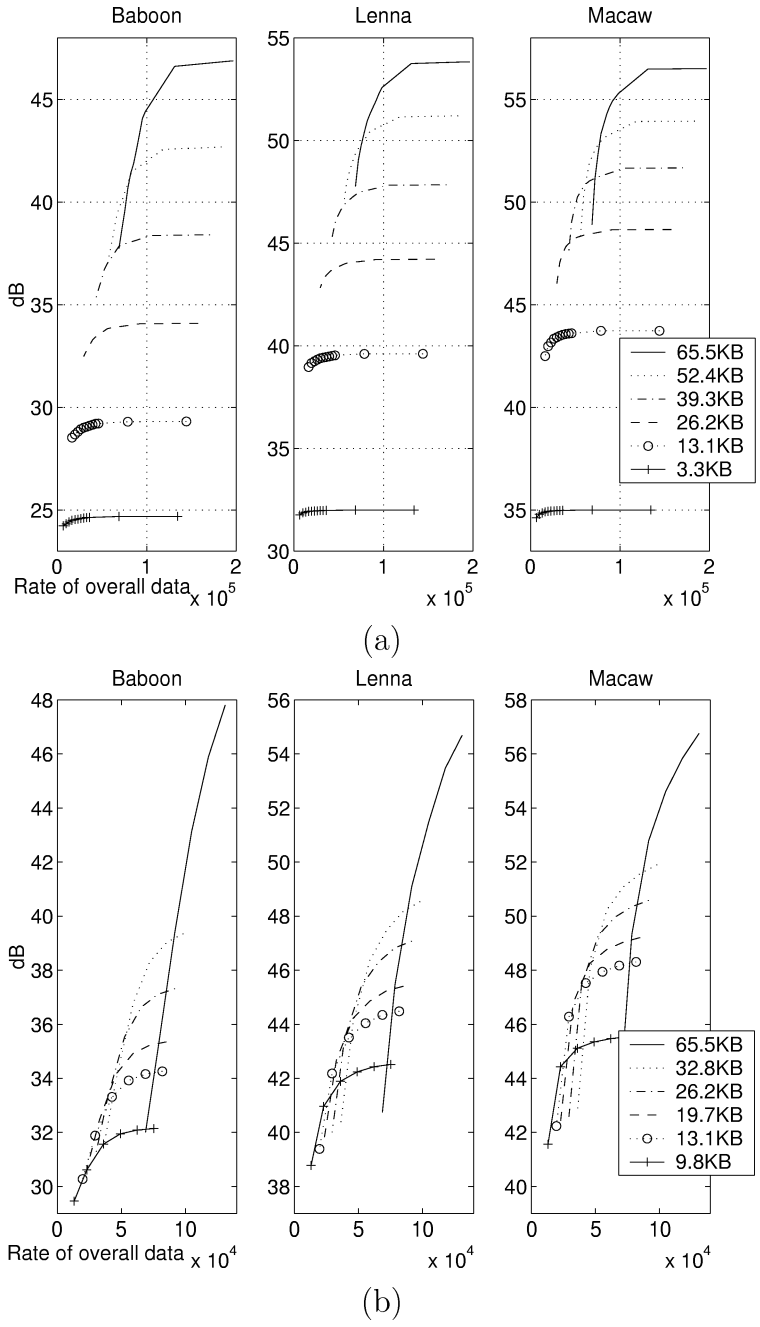


Figure 2.13: The curves indicate the PSNR of (a) luminance and (b) chrominance data after interpolation depending on the overall bit-rate. In (a), graphs are similar to those in Fig. 2.12 (a) (which use 2 by 2 block format conversion) except the horizontal axis (bit-rate of the overall compressed data). The bit-rates shown correspond to (a) luminance and (b) chrominance data.

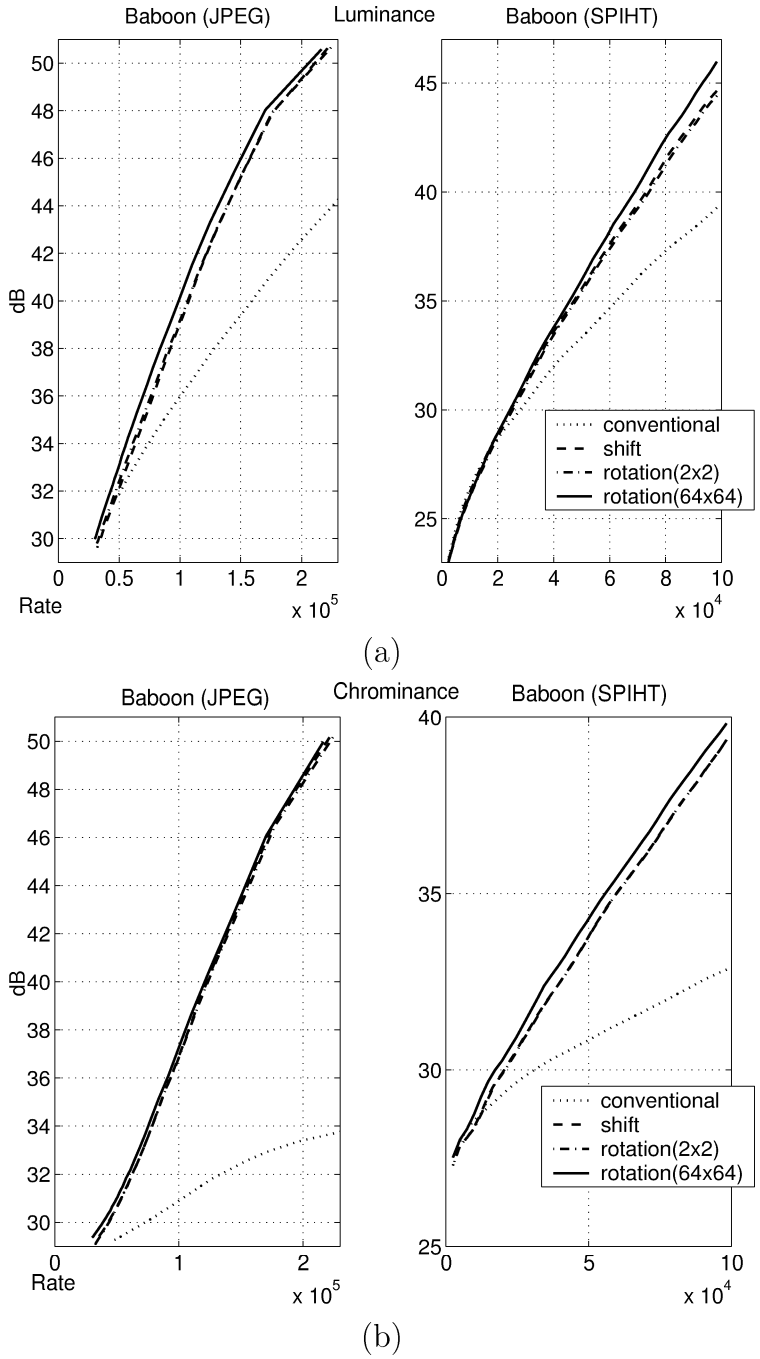
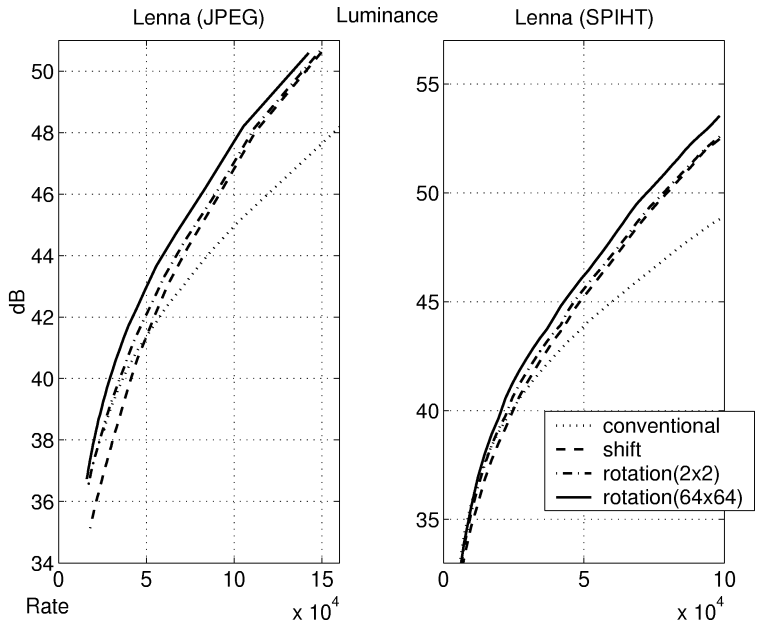
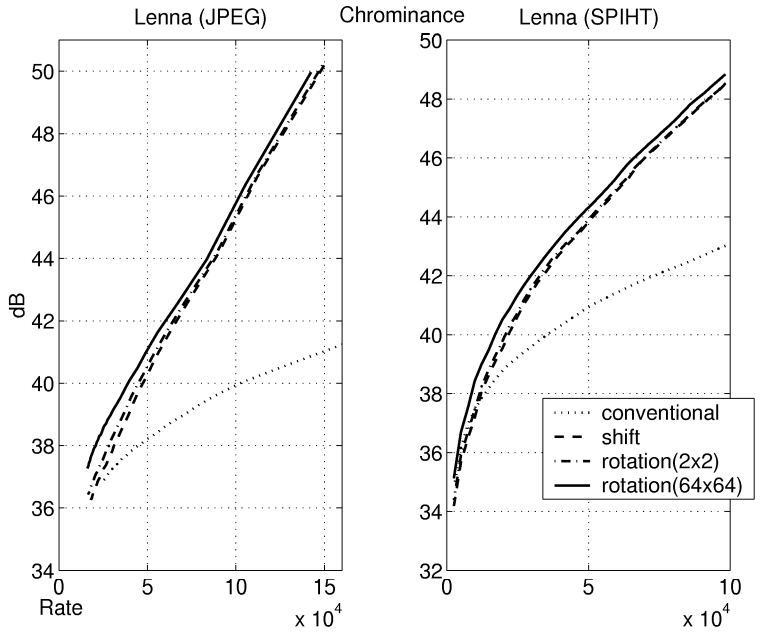


Figure 2.14: The curves indicate the luminance and chrominance PSNR after applying overall coding schemes.

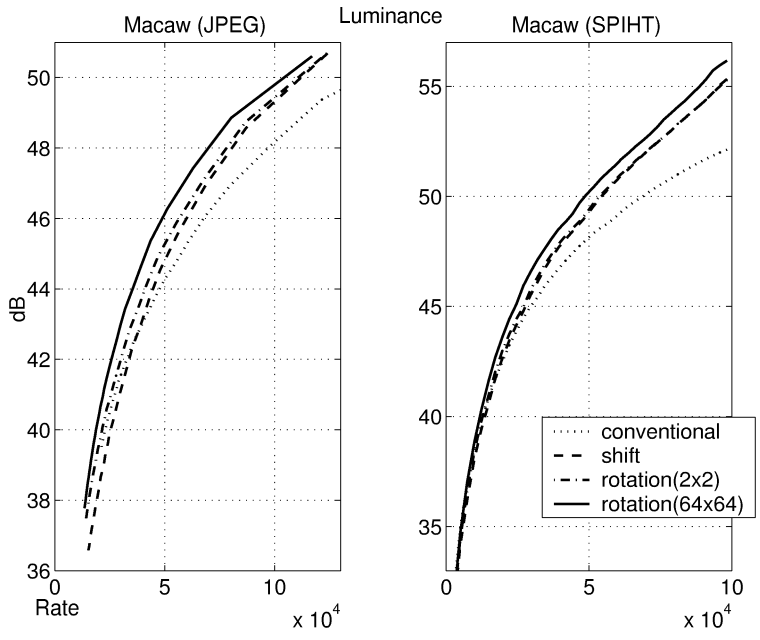


(c)

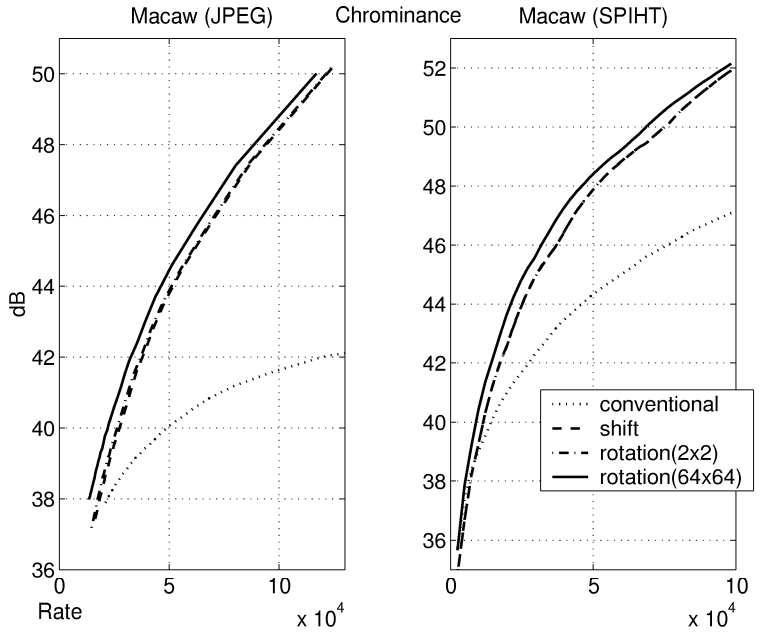


(d)

Figure 2.14 - continued



(e)



(f)

Figure 2.14 - continued

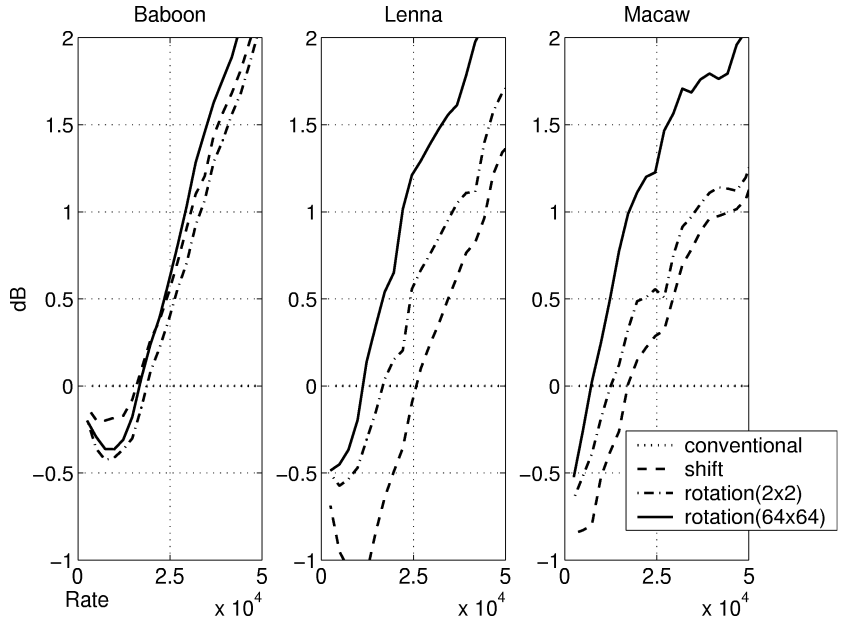


Figure 2.15: The PSNR gain of different proposed methods against the conventional method. Vertical and horizontal axes indicate the luminance PSNR gain and overall bit-rate respectively and SPIHT is used as a compression method.

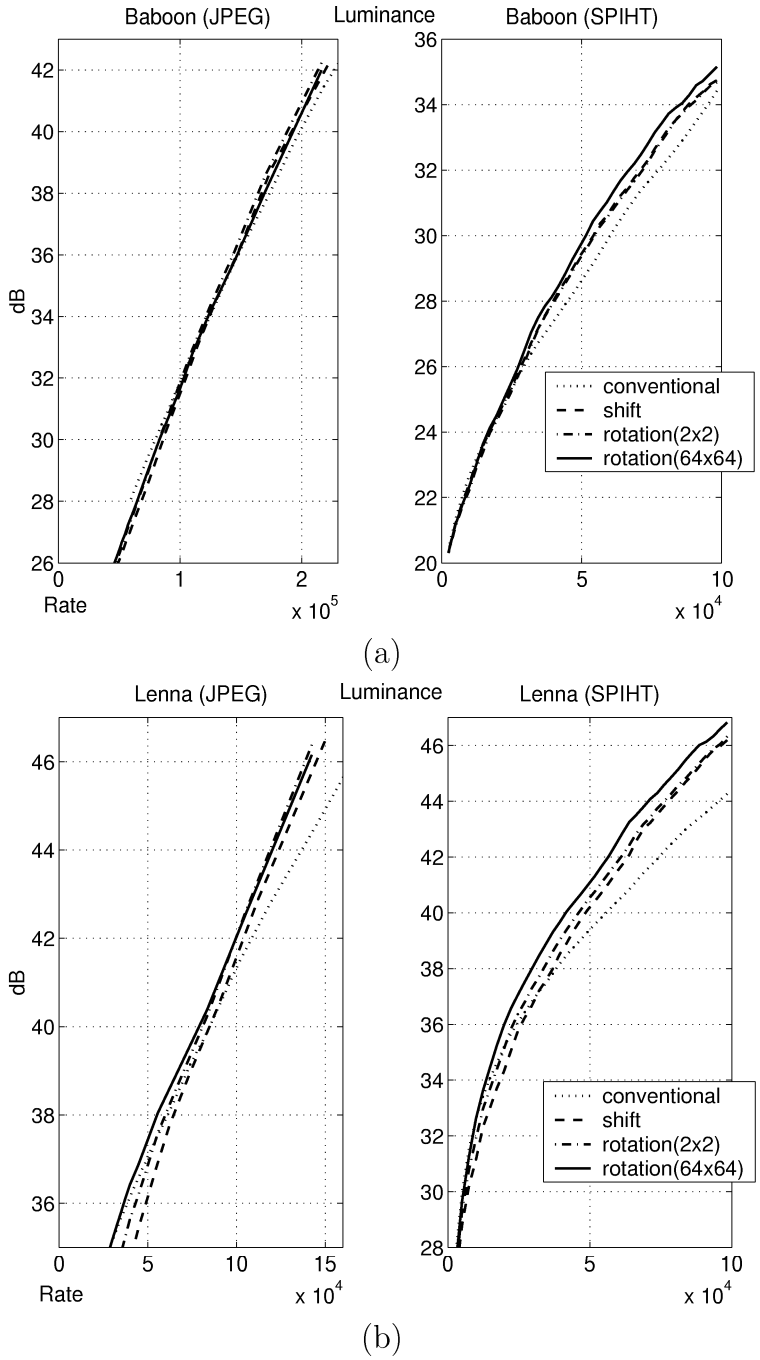
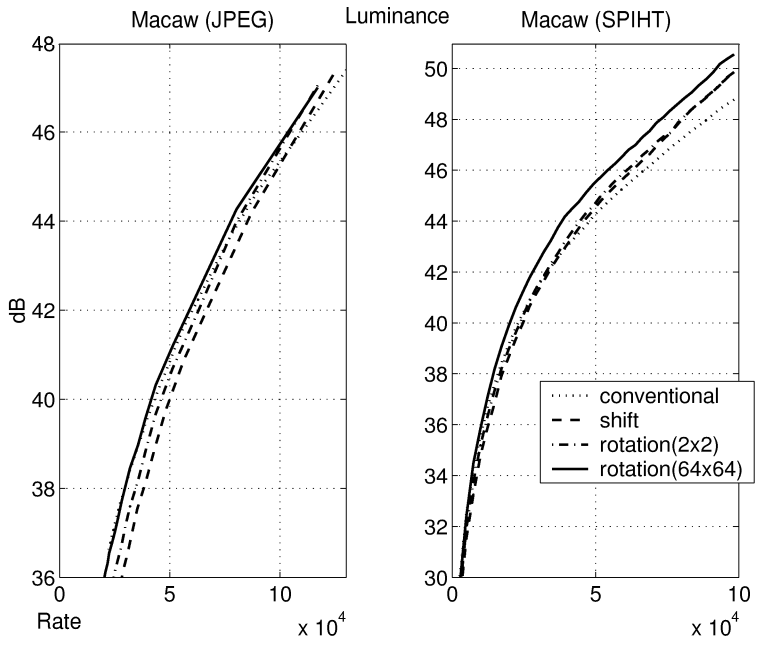
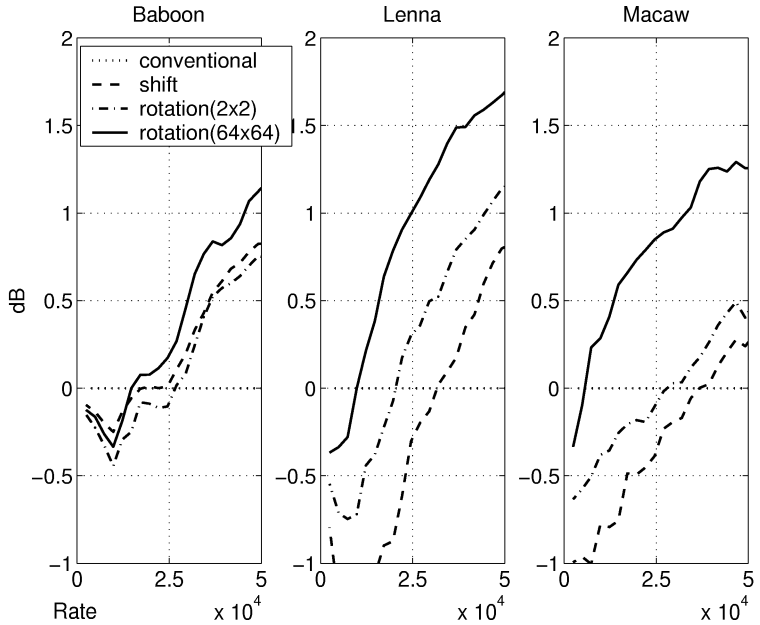


Figure 2.16: The curves in (a), (b) and (c) indicate the luminance PSNR after applying overall coding schemes with gradient based interpolation. The curve in (d) indicates the PSNR gain of different IAD methods against the CAI method with SPIHT.

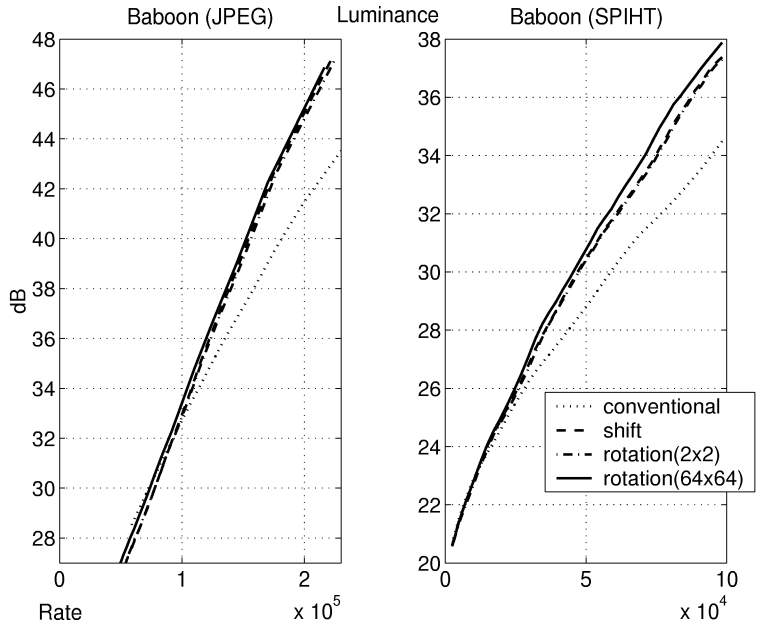


(c)

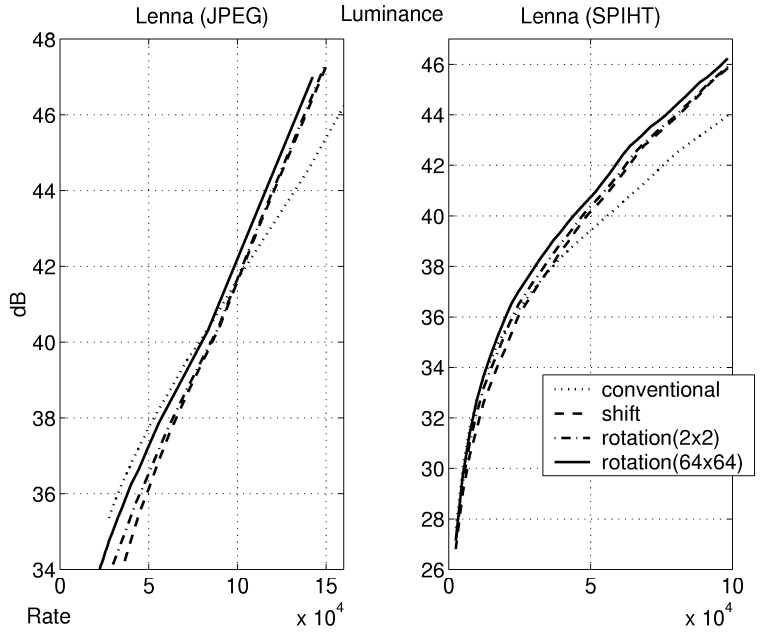


(d)

Figure 2.16 - continued

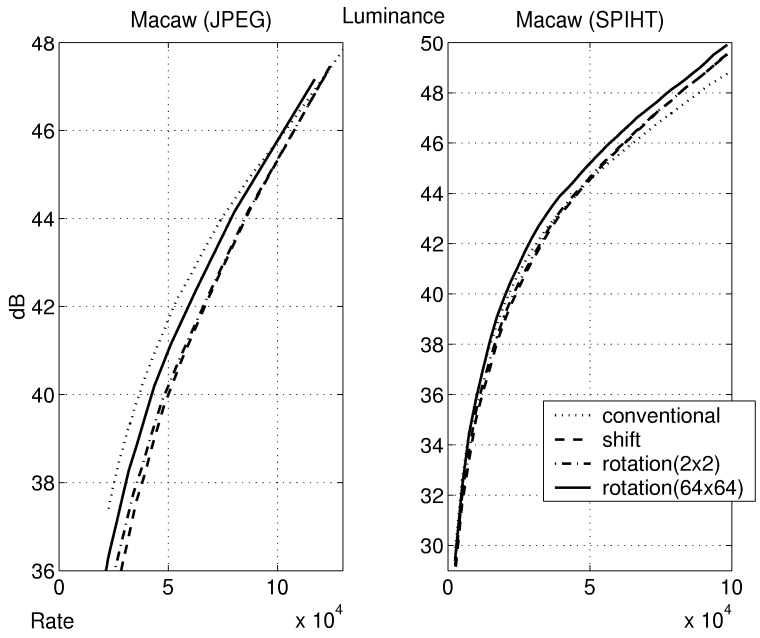


(a)

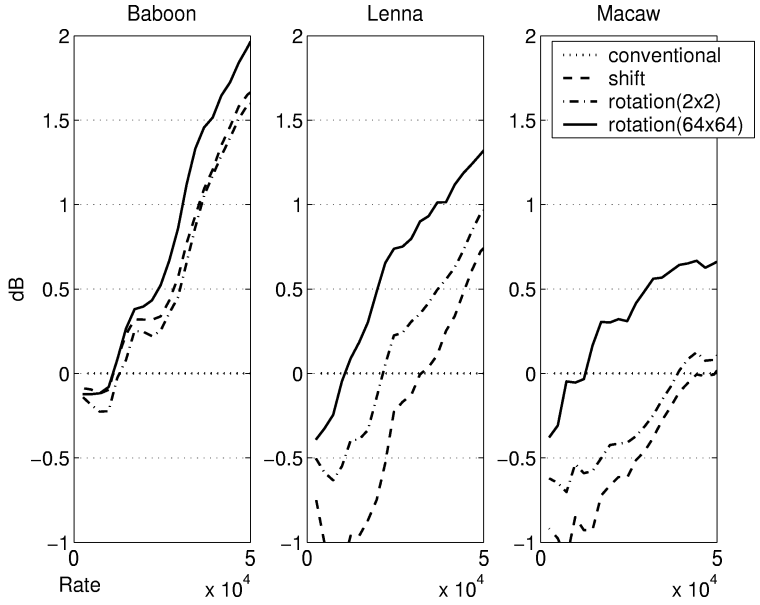


(b)

Figure 2.17: The curves in (a), (b) and (c) indicate the luminance PSNR after applying overall coding schemes with median-based interpolation and SPIHT. The curve in (d) indicates the PSNR gain of different IAD methods against the CAI method with SPIHT.



(c)



(d)

Figure 2.17 - continued

Chapter 3

Online Rate Control in Digital Cameras for Near-constant Distortion based on a MMAX criterion

In this chapter, we address the problem of online rate control in digital cameras, where the goal is to achieve near-constant distortion for all the images. In digital cameras, it is desirable to allow users to take a pre-determined number of images that can be stored within the given memory size, and captured/stored within a short delay, so that each image can be stored before the next image is received. Therefore, we need to define an online rate control that is based on the amount of memory used by previously stored images, the current image, and the estimated rate of future images. In this chapter, we propose an algorithm for online rate control, in which an adaptive reference, a "buffer-like" constraint, and a minimization of maximum distortion criterion (MMAX) are used in order to achieve

near-constant quality. The adaptive reference is used to estimate the R-D statistics of future images and the "buffer-like" constraint is required so that enough buffer space is preserved for future images. We show that using our algorithm to perform online bit allocation for each image in a randomly given set of images provides near constant quality. Also, we show that our result is near optimal when a MMAX criterion is used, i.e., it achieves a performance close to that obtained by applying an off-line optimal rate control that assumes exact knowledge of the images. Suboptimal behavior is only observed in situations where the distribution of images is not truly random (e.g., if most of the "complex" images are captured at the end of the sequence.) Finally, we propose a T-step delay rate control algorithm and using the result of 1-step delay rate control algorithm, we show that this algorithm removes the suboptimal behavior.

3.1 Introduction

Digital cameras are designed so as to mimic the operation of conventional cameras, which can take a pre-determined number of photos per roll. Thus for digital cameras it is desirable to enable the user to take a pre-determined number of images that can all be stored in the given memory size. Due to memory restrictions and time delay considerations, it is assumed that previously compressed images

will not be re-compressed and that images are compressed immediately after being captured and before the next image is received. Therefore, the bit allocation for digital cameras we consider here can be defined as an on-line bit allocation under constraints on the total memory and on the number of images. Sometimes, the photos taken by digital cameras are downloaded to other media such as a PC before the pre-fixed maximum number of images has been captured and stored. For this case, methods to improve the quality of the images using the remaining memory have been studied [17]. By using embedded quantizers, one image is saved twice in different memory areas and, during the decoding process, the quality of the output is improved by using both images. In this chapter, we assume that a pre-fixed maximum number of images is taken. That is, our goal is to determine the number of bits to use for each incoming image, given that the total number of images and the memory are fixed, and to do so in such a way as to provide as constant a quality as possible.

In order to solve this problem, we first need to select an appropriate distortion metric. As explained in section 1.2, there are three types of criteria that have been used for optimal bit-allocation, namely, minimization of average distortion (MMSE), minimization of maximum distortion (MMAX) and minimization of distortion under lexicographical constraints (MLEX) [47]. MMSE is by far the most popular criterion to evaluate the performance of image/video coding algorithms. As a consequence, optimal bit allocation under various constraints

for the MMSE criterion has been extensively studied in the literature. Examples include bit allocation for arbitrary inputs and a discrete set of available quantizers [64], bit allocation for dependent quantization [51], and optimal bit allocation under buffering constraint [48]. Although the MMSE criterion gives smallest total distortion for a given budget and efficient algorithms are available, it does not guarantee the constant or near-constant level of distortion that may be more important for human viewers, while the MMAX and MLEX criteria are better suited for this purpose [58]. Solutions for the bit allocation problem under the MMAX criterion for both independent [41] and dependent quantizers [60] have been studied. The MLEX criterion has been proposed as an extension of MMAX [20]. This criterion involves minimizing the maximum quantization value and then minimizing the second highest quantization value and so on, while the MMAX criterion aims to minimize the maximum distortion. Based on the MLEX criterion, real-time VBR rate control for MPEG video has been studied [19].

In a digital camera application, because each image will be viewed independently, it is desirable to provide very consistent quality, so that no image appears to have significantly worse quality than others. In terms of bit allocation, this means that bits should be allocated iteratively to reduce the distortion of the worst (highest distortion) image. From this perspective, for digital cameras, either the MMAX or the MLEX criteria are more suitable than the MMSE criterion. Moreover, unlike video encoders, we do not need to consider buffering

constraints because there is not strict time restriction for decoding consecutive images stored in digital cameras, while a video sequence should be decoded for being displayed at a certain frame rate.

Therefore, if we knew in advance all the images to be stored in the camera then the bit-allocation problem would be exactly the same as that of finding the optimal bit allocation given a set of discrete quantizers for either the MMAX or MLEX criterion. Moreover in an off-line optimal bit-allocation scenario, due to the lack of buffering constraints, and the independent quantization of the images, the solution is relatively straightforward. For example, under the MMAX criterion, the problem is equivalent to finding the minimum bit rate under a given maximum distortion constraint,

$$R^*(D_{max}) = \min_{x_1,\cdots,x_N} R(x_1,\cdots,x_N), s.t. : D(x_1,\cdots,x_N) \leq D_{max} \tag{3.1}$$

and then solving this problem iteratively with lower maximum distortion constraints until the available bit budget has been exhausted [58].

Clearly, in real-world scenarios the complete set of images is not available and thus we are faced with the task of making decisions on incoming images, without any knowledge of future images, and still trying to meet a MMAX optimization criterion. Moreover, we cannot change the allocation for those images that have already been coded.

In this chapter, we address the problem of online rate control in digital cameras. In particular we provide a "buffer-like" constraint to keep enough memory for future images. Our goal is to determine the online bit allocation for each image such that near constant quality is achieved by using a MMAX optimization criterion without violating given restrictions such as the fixed size of memory and the pre-determined number of images. We show that the result of the proposed online method is better than that of other methods such as constant rate, constant quantizer, and constant distortion.

Under the assumption that a larger size working memory may be available in some cases, we propose a T-step delay rate control algorithm that uses the information of T known future images for compressing a current image. (Here, the current image is the image to be compressed and known future images are images that are in a working memory and will be compressed after the current image.) We also show that the result of the proposed 1-step delay method (i.e., there is one known future image) leads to much better results the online method without delay.

This chapter is organized as follows: in section 2 an online bit allocation algorithm is presented. Experimental results are provided as demonstration of the validity of our algorithm in section 3. Finally, the conclusion of this work is in section 4.

3.2 Online bit allocation

In order to determine the desirable distortion for a particular image that will guarantee constant or near-constant distortion for all images, we first need to estimate the rate-distortion (R-D) characteristics of future images. In video coding it is reasonable to expect that images within a video scene will have similar R-D characteristics, and thus the current image characteristics are a good indication of those of the next image. However no such assumptions can be made in a digital camera environment. Instead, we can assume that the future images may have R-D characteristics that correspond to the average of all images taken so far by the digital camera. Under the assumption that the size of the image is same, we can consider that the R-D characteristics of the images, e.g., the distortion for a given rate, are independent identically distributed random variables. So, if the number of images is large enough then, by the weak low of large numbers, the mean value can be determined by the average of all images [9] [69]. Therefore the average R-D characteristics is determined by the average rate of the images that have been captured under given distortion by the current image. In the discrete quantizer case, the distortion determined by each quantizer index is different for each image. Therefore, an interpolation method is needed to get the average bit-rate of a given distortion.

Obviously, this estimation is not accurate in general and, as a result, this can lead us to coding images with a distortion below what would be desirable. Assume for example that in a set of captured images, the most complex images (i.e., those for which high quality requires high rate) are captured first and the simplest images (high quality can be achieved with lower rate) are captured towards the end. Then, if we only use information about past images, an optimal bit allocation cannot be achieved, because we will overestimate the complexity of future images and therefore allocate fewer bits than necessary to the initial ones.

We do not expect these situations to occur often, under the assumption that image complexity can be roughly random for a given set of images. However, we do define some additional constraints for our allocation problem to prevent that situations such as the one described above have a negative impact on quality. Consider the extreme case of a high complexity image. For high complexity images, let R^h_{min} be the minimum rate that has to be provided in order to achieve the minimum acceptable quality, which can be determined by a visual test. If R^h_{min} is the same as the average rate of each image (i.e., higher quality is guaranteed for high complexity images), then the distortion is always better than that of a constant rate method (since the guaranteed bit-rate for high complexity image is at least the bit-rate used in the constant rate method). But the flexibility of bit-rate of each image is reduced since we need to reserve more bits for future use and so the resulting distortion has large variation. Conversely, if R^h_{min} is very

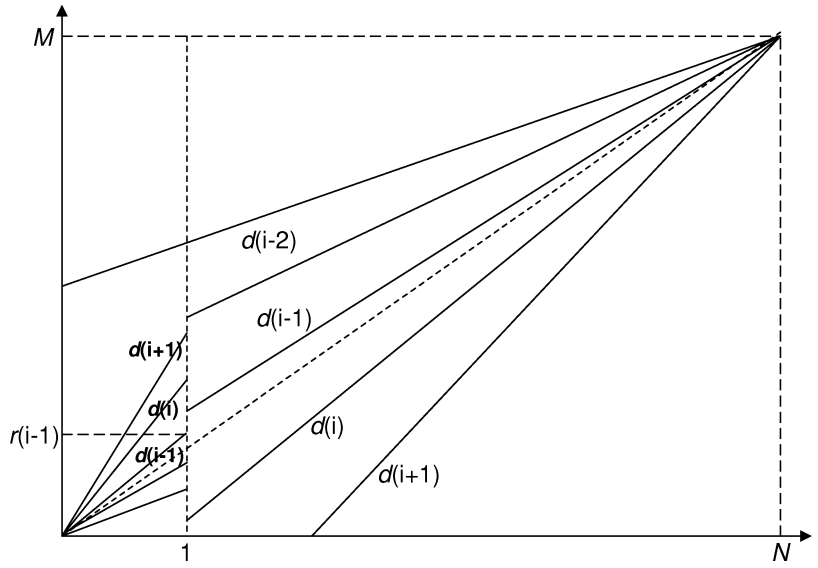


Figure 3.1: The lines from the lower left corner indicate the memory occupation of the first image determined by given quantizers. The lines from the upper right corner indicate the average memory occupation by $N-1$ unknown images with the same distortion. The solution is the line that has minimum distance between two lines under the same distortion. (In this case, $d(i-1)$ is the solution of the first image.)

small, then there is enough memory to keep the distortion constant, but if the solution meet R^h_{min}, then the quality of the image may not be acceptable.

By using R^h_{min}, we can define a useful bound. Refer to Fig. 3.1, where we assume the first image in a set is coded and there are $N-1$ images remaining. Without any additional knowledge aside from R^h_{min} we can then introduce the following constraints. The worst case scenario is that all the future images are high complexity, and we would like to guarantee that the minimum acceptable quality can be achieved for all future images even in that case. Thus, when coding

image k we need to ensure that there are at least $(N-k)\cdot R^h_{min}$ bits remaining. This is represented by the top solid line in Fig. 3.1

With the above "buffer-like" constraint we have imposed a constraint that would tend to avoid poor quality under worst case circumstances (i.e., in case that the complexity of images is not random). We now explain our online algorithm to ensure near constant quality under this constraint.

Let N be the number of images to be stored and M be the size of the memory. In Fig. 3.1, each of the lines that start from the lower left corner (LLL) correspond to the bit rate of image 1 under each available quantizer. The lines that start from the upper right corner (LUR) represent the estimated bit rates of the future images under the assumption that they all have the same characteristics as the first image. Each line again corresponds to a given quantization choice. For example, $d(i)$ indicates the distortion of image 1 under the i^{th} quantizer and $r(i)$ indicates the bit rate under the same quantizer where large quantization index means lower distortion. Because the LUR are obtained by subtracting allocated rate from the total rate the rate for each image from image N to image 1, the order of the LUR is opposite to the LLL. As an example, if quantizer i requires a large number of bits then the LUR corresponding to this quantizer will be low, since at frame k this line indicates rate available for the first k frames, assuming the last $N-k$ use quantizer i.

After capturing one image, we can assume that the image sequence will contain this image and the $N-1$ images that have the average R-D characteristics. Refer again to Fig. 3.1, where we plot for image 1 all possible quantizer allocations and their corresponding distortion. Then for the remaining $N-1$ images we plot the bit rate demand if all images were coded with the same quantization scale and had the average R-D characteristics. For example, if all future images were coded with quantizer i we would not have enough memory to accommodate them (assuming they all have average R-D characteristics.) Conversely, if quantizer $i-2$ is used, sufficient rate is available for all remaining images. Our goal then is to choose, given the expected R-D characteristics for the future images, a quantizer for image 1 such that (i) the upper bound is not violated, (ii) the difference between the quality of current image and future images is minimized. This second condition is aimed at achieving near constant quality in the set of images. If we have an infinite number of quantizer choices then one of the LLL can meet one of LUR that has the same distortion. But, in the discrete quantizer case, this usually does not happen. Therefore, for a discrete set of quantizers, the optimal solution would be to choose the quantizer i such that the lines (LLL and LUR) are closest to each other at frame k, with the LUR being above the LLL. Note that each of the algorithm, for frame k, the $k-1$ first frames are assumed fixed, and the LLL are started from the current buffer position (see Fig. 3.2). For example, in Fig. 3.1 if all future images are coded with quantizer i we would

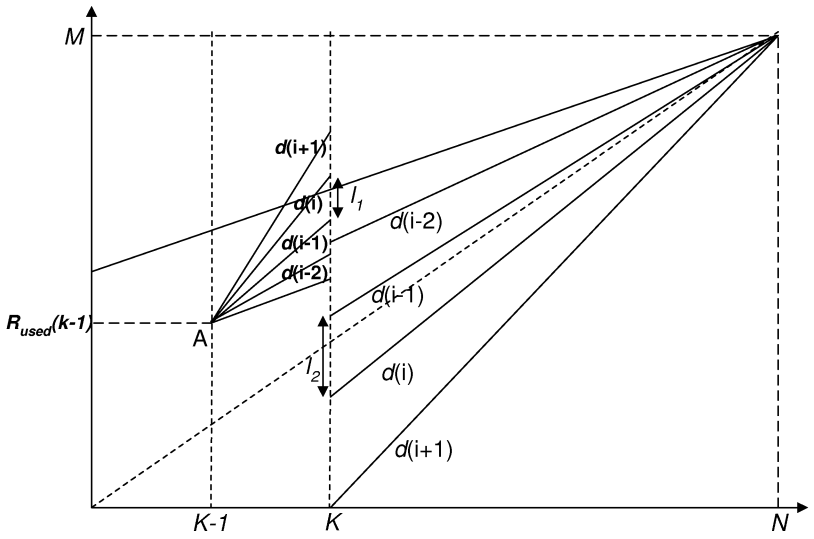


Figure 3.2: This graph shows the change of the solution depending on the bit allocation to the previous images. The lines originating from point A indicate the total memory occupied by the first k images for each quantization choice for image k. The lines from the upper right corner indicate the average memory occupation for $N-k$ unknown images with the same average R-D characteristics. The solution is the line that has minimum distance between two lines under the same distortion. (In this case, $d(i-2)$ is the solution for the k^{th} image even though the image is same as the first image in Fig. 3.1 and the average R-D characteristics in Fig. 3.1 is used.) $R_{used}(k-1)$ indicates the total memory allocated to the first $k-1$ images.

not have enough memory to accommodate them (assuming they all have average R-D characteristics.). So, in this example, the optimal solution is $d(i-1)$.

More formally, our problem to find a quantization value

$$q_k = \arg\min_{q_{ki}} \mid r(q_{ki}) + (N-k) \times R(d(q_{ki})) - M_k \mid \tag{3.2}$$

where k is the index of the image, i is the quantization index, $r(q_{ki})$ is the bit rate of the given q_{ki}, $d(q_{ki})$ is the distortion of the given q_{ki}, M_k denotes the

remaining bits at the k^{th} image and $R(d(q_{ki}))$ is the average bit rate for the future images determined by $d(q_{ki})$. If there is more than one solution then the solution that is nearer to the constant line (the dotted line in Fig. 3.1) is chosen for a current image and future images.

Refer to Fig. 3.2. Here the remaining memory is smaller than the average and this remaining memory should be shared by the k^{th} image and the $N-k$ future images. Therefore the optimal solution can be changed even if we use the same average R-D characteristics for all images (i.e., the R-D characteristics of previously stored images are not used to refine our estimation of the average R-D characteristics) and the R-D characteristics of the k^{th} image is same as that of the first image in Fig. 3.1. Although the optimal solution can be changed by the memory occupation of the previously stored images, this rate control based on the current utilization memory may be too slow and not allow us to keep the memory state under the given upper bound. The reason is that the distance between the rates corresponding to consecutive distortions for future images (e.g. l_2) is much larger than the distance between the rates of consecutive quantizers for the current image (e.g. l_1). For example, the rates difference of the future images are $N-1$ times larger than that of image 1 (if image 1 has the average R-D characteristics) since this difference is linear with the number of future images. Therefore, in the beginning part of the image sequence, the solution is determined dominantly by the average R-D characteristics for the future images. In order

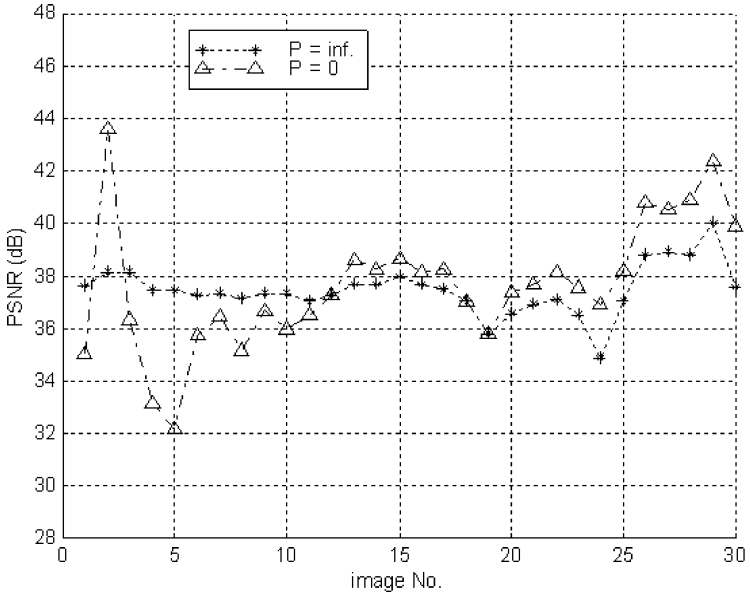
to keep the memory state under the given upper bound, we need to update the average R-D characteristics with each incoming image as

$$R(D_i, k) = \frac{(P + k - 1) \times R(D_i, k - 1) + r(D_i)}{P + k} \tag{3.3}$$

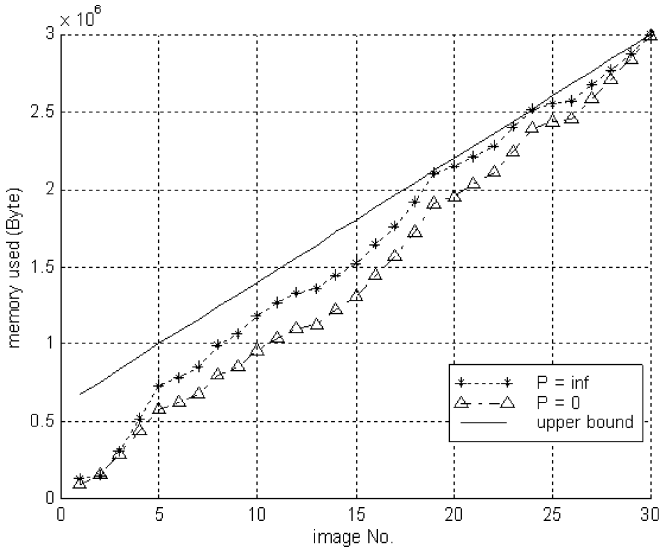
where $R(D_i, k)$ is the average R-D characteristics for the k^{th} image, D_i is the i^{th} distortion in a certain resolution, P is the weighting for the training result. If P is too small then the average R-D characteristics changes very fast and, as a result, the distortion of the image set can suffer substantial fluctuation. Conversely, P is too large then the result is same as that of the rate control based on a predetermined and fixed R-D model for all future frames. This is illustrated by Fig. 3.3, where two extreme cases of selection of P are shown. In $P = 0$ case, the PSNR fluctuates significantly but the memory state remains well below the upper bound converely, in $P = \infty$, the PSNR is flat but the memory state reaches the upper bound twice.

Although the probability of reaching the upper bound is reduced by the above updating method for the average R-D characteristics, still, there are cases where the upper bound is reached. In those cases, we should restrict the rate of the current image in order to save enough memory for future images.

Algorithm 1: *Online rate control for an image set.*



(a)



(b)

Figure 3.3: $P = \infty$ indicates on-line rate control with fixed average R-D characteristics and $P = 0$ indicates on-line rate control without average R-D characteristics given by pre-training (i.e., the average R-D characteristics is only determined by previously saved images and the current image). This image set is the same as the image set 2 in Fig. 3.5.

[Step 0]: *Find average rate distortion characteristics ($R(D)$), using the training set of images.*

[Step 1]: *Update the average rate distortion with the rate distortion of current image using (3.3).*

[Step 2]: *For the k^{th} image, find the quantization value q_k using (3.2). If the solution violates the boundary condition then, find q_k using,*

$$q_k = \arg\min_{q_{ki}} \left(\frac{N - k \times R(d(q_{ki})) + r(q_{ki})}{N - k + 1} - R^h_{min} \right)$$

subject to

$$\left(\frac{N - k \times R(d(q_{ki})) + r(q_{ki})}{N - k + 1} \right) \geq R^h_{min}.$$

[Step 3]: *If the current image is the final image then end, else $k = k + 1$, goto* ***Step 1***.

More generally, we assume that we have not only R-D information for the current image but also R-D information for the next T future images (i.e., we can store $T + 1$ images before compression.) Under this assumption, the T known future images can be used for updating the average R-D characteristics and for selecting the optimal bit allocation for the current image. In this T-step delay rate control algorithm, the initial average R-D characteristics is obtained from the

training sets and the first T images in the image sequence and then it is updated by incorporating newly updated images. Therefore, if $T = N$ and $P = 0$ (as an extreme case), then we can use off-line MMAX optimal R-D characteristics of the image set for the whole images. The new advantages of the T-step delay algorithm are that (i) the effect of the known future images is taken into account to make decisions on the current image and (ii) the upper bound can be ignored if the memory state is below the upper boundary after T images. The memory state of a future image can reach the upper bound when the online rate control is used. However, by using the T-step algorithm, the memory state of the known future images can be kept below the upper bound by reducing the rate of the current image. For example, in the online rate control case, if the optimal solution of the current image violates the upper bound constraint then the solution should be changed for the future images. But in T-step delay rate control case, if the optimal solution of the current image and T known future images satisfies the boundary constraint then the optimal solution is selected as the solution of the current image, even if this solution violates the constraint after storing the current image. Therefore, using this second advantage, checking that the upper bound is not violated only needs to be done with the result after T-steps (and not with the result at every step) in the Algorithm 2 Step 3.

The main difference between the online and T-step delay rate control algorithms is that the T-step delay rate control uses (i) in Step 1, the average R-D

characteristics is updated with $T+1$ images before compressing the first image and then with each new image and (ii) in Step 3, the R-D characteristics of T-step ahead images are used to determine the solution of the current image.

Algorithm 2: *T-step delay rate control for an image set.*

[Step 0]: *Find the average rate distortion ($R(D)$), using the training sets of images.*

[Step 1]: *Iterate (3.3) T times with increase of k and then reset $k = 1$.*

[Step 2]: *Update the average rate distortion with the rate distortion of the next image captured, i.e., the T^{th} image captured after the current image.*

$$R(D_i, k) = \frac{(P + k + T - 1) \times R(D_i, k-1) + r_{k+T}(D_i)}{P + k + T}; T < N - k$$

where $r_{k+T}(D_i)$ is the rate of the $(k+T)^{th}$ image at a distortion D_i.

*If $T \geq N - k$, then skip **Step 2**.*

[Step 3]: *For the k^{th} image, find the quantization value q_k as*

$$q_k = \arg\min_{q_{ki}} \mid r(q_{ki}) + \sum_{j=1}^{T} r_j(d(q_{ki})) + (N - T - k) \times R(d(q_{ki})) - M_k \mid$$

If the solution violates the boundary condition then find q_k using,

$$q_k = \arg\min_{q_{ki}} \left(\frac{(N-T-k) \times R(d(q_{ki})) + r(q_{ki}) + \sum_{j=1}^{T} r_j(d(q_{ki}))}{N-k+1} - R_{min}^h \right)$$

subject to

$$\left(\frac{(N-T-k) \times R(d(q_{ki})) + r(q_{ki}) + \sum_{j=1}^{T} r_j(d(q_{ki}))}{N-k+1} \right) \geq R_{min}^h.$$

[Step 4]: *If the current image is the final image then end, else $k = k+1$. If $T \geq N-k$ then T=T-1. Goto* ***Step 2****.*

3.3 Experimental results and discussion

In order to confirm the validity of the proposed online algorithm, we implement this algorithm and test it with 285 images taken from MPEG7 test images (H: 512, W: 768 or vice versa). In the implementation, we assume that the total memory size available is 3 Mbytes and the pre-determined number of images is 30 (the total number of images in one image set), i.e., the average bit rate of each image is 100 Kbytes. 70 image sets are randomly generated from the test images for training the encoder to get the average R-D characteristics. In Table 3.1, the results of the proposed online and 1-step delay methods are compared with other

methods. The results are obtained by averaging those obtained with 30 image sets randomly generated from the image set. Note that we used all test images to generate both training sets and test sets, but under the assumption that the images are randomly distributed, the main result of the training depends on the order of the randomly selected images. Therefore our experiment is reasonable because the training sets and the test sets are different and the test images are randomly distributed. The variation in image charateristics is shown by the bit rate change under given distortion in Figs. 3.4(c) and 3.5(c)). The results of the proposed algorithms are better than the constant rate and constant quantization algorithms in terms of average minimum PSNR and of average standard deviation of PSNR. Because for high complexity images more bits are needed to decrease the distortion than for low complexity images, the average PSNR of the off-line MMAX optimal solution is worse than for the other methods. Here, for the constant quantization and constant distortion methods, we select the quantizer index and the distortion based on the training result. For better results, a larger quantizer index (i.e, a smaller quantization step size) and smaller distortion can be selected but this induces more frequent violations of the memory constraint. The last column shows the number of saved images. This shows that the constant quantization and constant distortion methods cannot guarantee storing the predetermined number of images in a fixed memory size.

The results for two specific image sets are given in Figs. 3.4 and 3.5. In Figs. 3.4(c) and 3.5(c), we can see that the complexities of the image sources are randomly distributed and the high complexity images need approximately 10 times higher bit rate than the low complexity images for a given distortion.

In Figs. 3.4 and 3.5, the results of the proposed online algorithm are compared with several methods such as off-line MMAX optimization, constant rate, constant quantization, and constant distortion using randomly selected image sets. For the constant rate method, we select the quantization value which satisfies the condition that the compressed image size is under 100 Kbytes, in order to prevent violations of the total memory size. From Figs. 3.4(b) and 3.5(b), we can see the PSNR curve of the constant quantization method that is used dominantly in digital cameras produces fluctuations of more than 10dB. Although the PSNR curve of constant distortion looks near optimal in Fig. 3.5(b), the used memory is over 3 Mbytes (i.e., we can not save all the pre-determined number of images.). Also, in Fig. 3.4(b), the PSNR curve is below the PSNR curve of the online method due to over estimation. (We can see that more than 20% of the memory remains unused in Fig. 3.4(f).) As we already mentioned, for the set of images that contains many high complexity images, we cannot save the pre-determined number of images using either the constant quantization or the constant distortion method. Although the constant rate method can always save

the pre-determined number of images, this method gives the worst result in terms of providing a constant distortion.

From both figures, we can verify that the online algorithm gives better results than the other three online methods without resulting in memory violations. When the image set is not random (i.e., several high complexity images are captured successively), this algorithm shows sub-optimal behavior by the "buffer-like" constraint. But, with the help of this constraint, we can always store the pre-fixed number of images with, at least, the minimum acceptable quality.

In Fig. 3.6, the results of the proposed online and 1-step delay methods are compared with those obtained with the off-line MMAX optimal method using two image sets. Fig. 3.6(a) shows that the result of the proposed online method tends to have low PSNR due to the effect of the upper bound. Because this image set has many high complexity images (the off-line optimal PSNR is under 37dB), the average R-D characteristics determined by the training sets clearly underestimates the complexity of the images actually being coded. Due to this mismatch, the memory state reaches the upper bound and tends to low PSNR result. The 1-step delay method eliminates this problem by reducing the rate of the previous image. Thus, the average result of the 1 step delay method is much better than the online method (see Table 3.1).

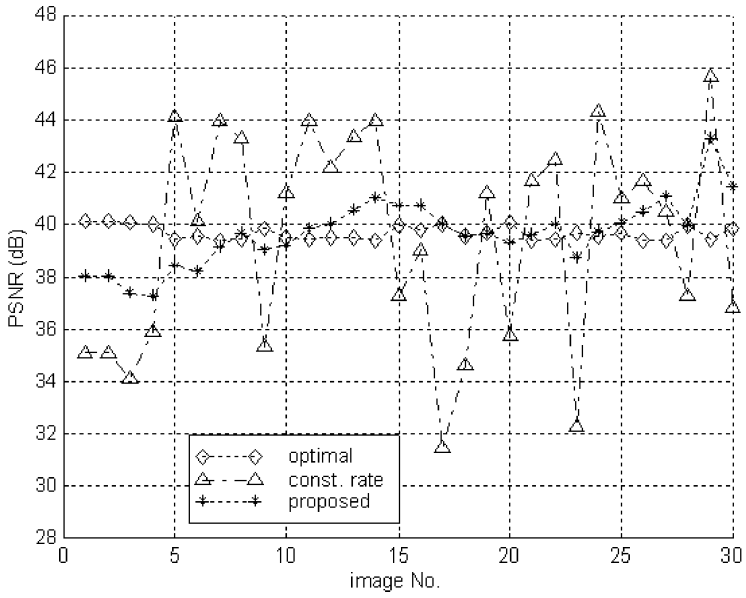
Table 3.1: Average performance (PSNR) comparison of proposed online algorithm with off-line optimization, constant rate, and constant quantization using 30 image sets composed of randomly chosen 30 images. The last column indicates total number of saved images out of 900 images.

Method	Average	Std. Dev.	Minimum	Maximum	Number of saved images
Off-line Optimal	37.98	0.173	37.79	38.42	900
Constant Rate	38.01	4.22	28.56	45.80	900
Constant Quantizer	38.17	2.250	34.17	44.78	879
Constant Distortion	37.82	0.139	37.48	38.12	872
Proposed(Online)	38.10	1.546	34.55	41.83	900
Proposed (1 step delay)	38.02	0.413	35.62	40.81	900

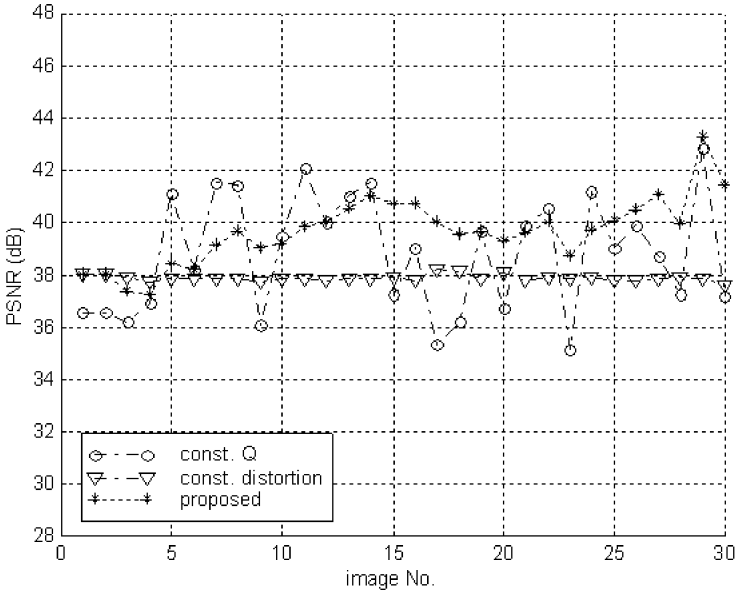
3.4 Conclusions

In this chapter, we investigated the optimal bit allocation problem in digital cameras, i.e., the problem of determining the optimal bit rate for a current image without having information about the future images, under the constraint of storing a fixed number of images, within a fixed memory size. The characteristics of future images are estimated from the training data and the images already taken and the problem can be formulated as minimizing the maximum distortion for a current image and the estimated future images. In order to keep enough memory for the future images, we used a "buffer-like" constraint. We showed that our result achieved a performance close to that obtained by applying an off-line rate control.

We also investigated the optimal bit allocation problem under the assumption that T future images are available. With this limited number of known future images, suboptimal behaviors arising in the no look-ahead problem can be eliminated. We showed that even the 1-step delay rate control method had much better results than the online method with no look-ahead while keeping the memory state below the upper bound.

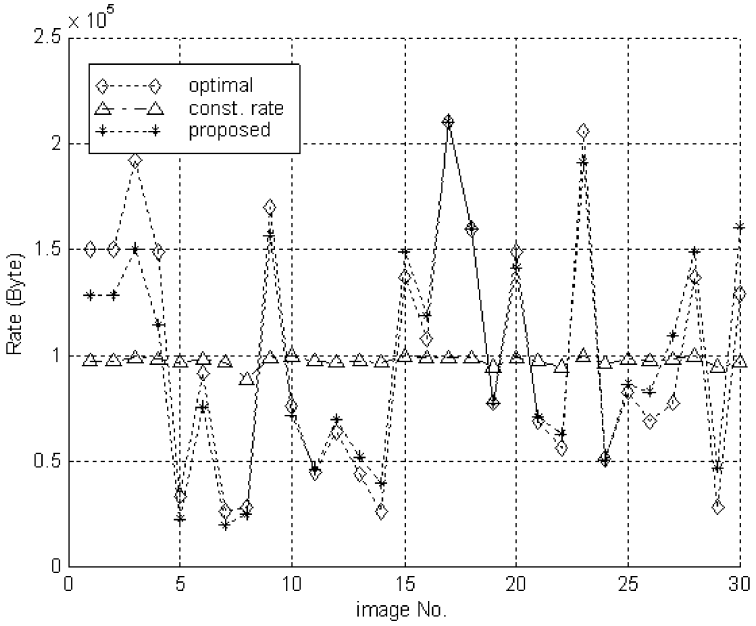


(a)

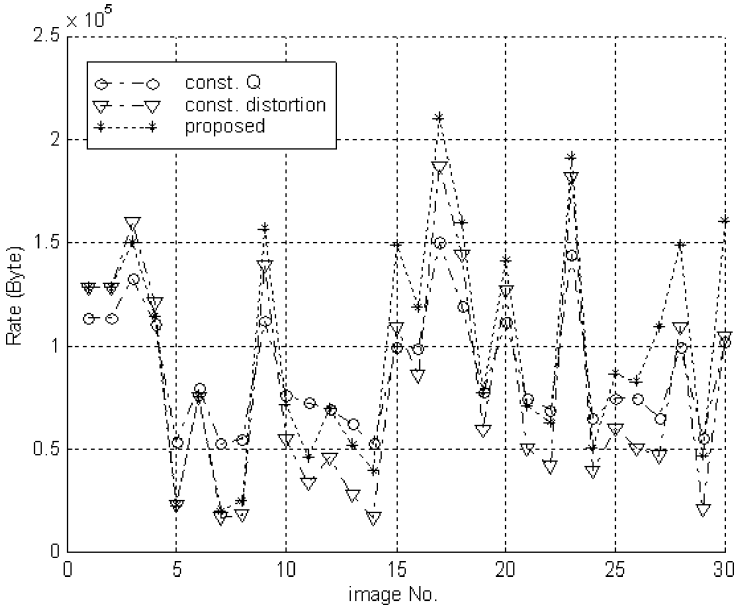


(b)

Figure 3.4: Performance comparison of proposed online algorithm with other methods such as off-line optimization, constant rate, constant quantization, and constant distortion using image set 1 composed of randomly chosen 30 images: (a), (b) PSNR of each image, (c), (d) bit rate of each image and (e), (f) memory usage for this image set.

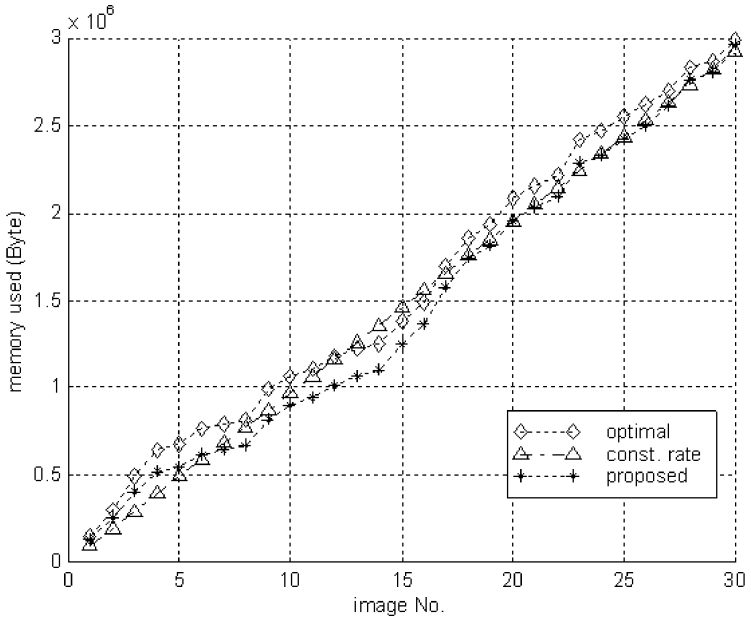


(c)

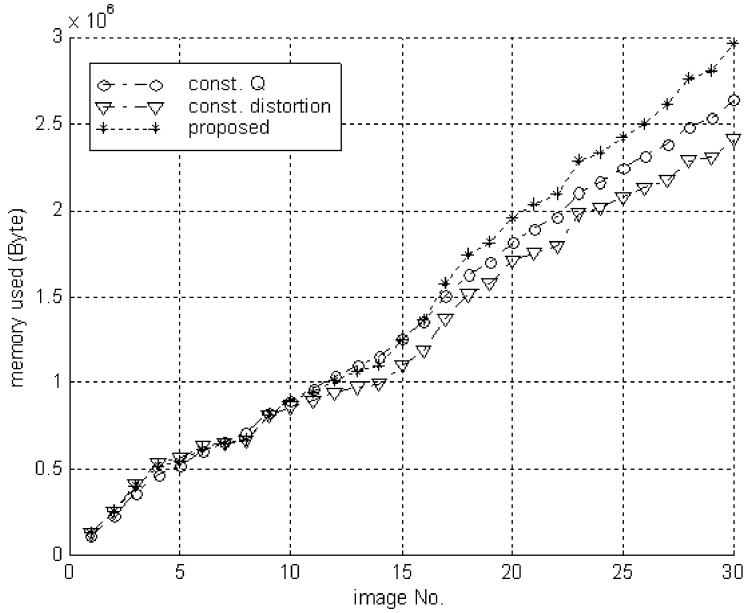


(d)

Figure 3.4 - continued

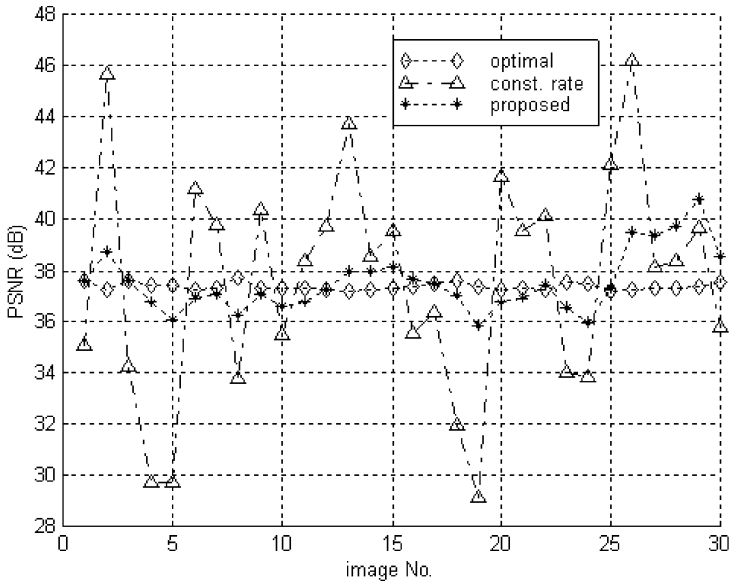


(e)

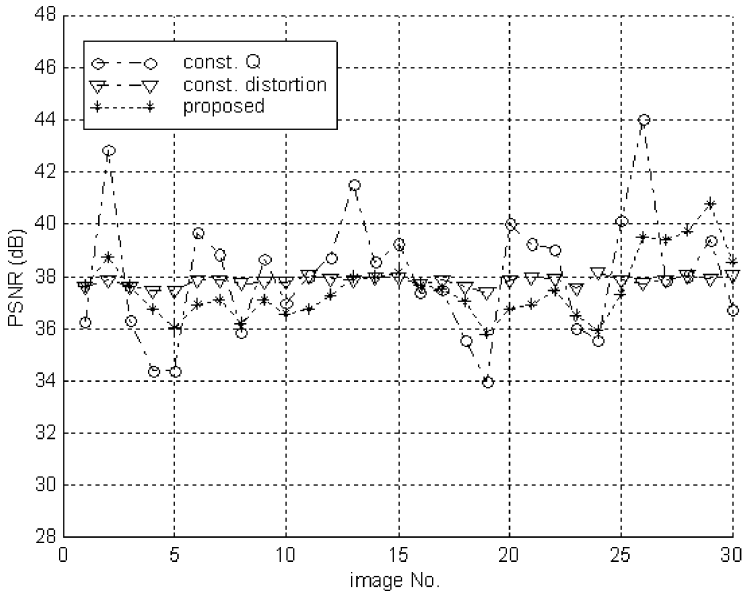


(f)

Figure 3.4 - continued

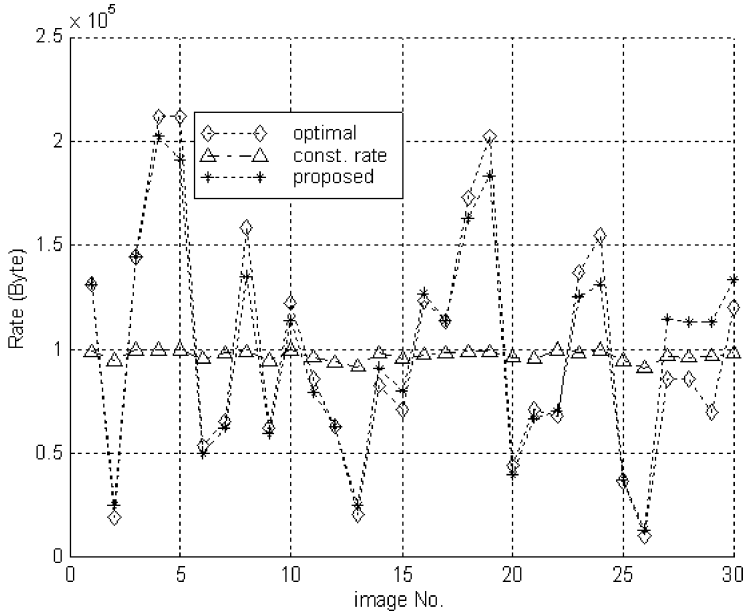


(a)

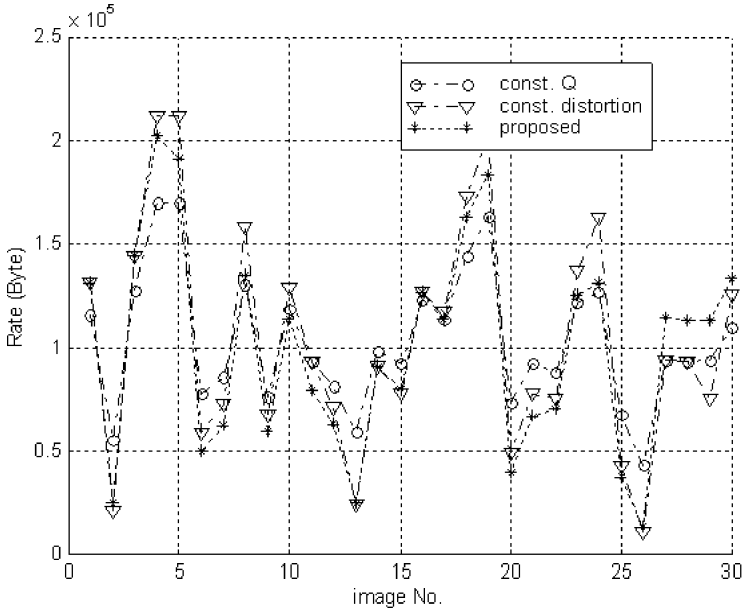


(b)

Figure 3.5: Performance comparison of proposed online algorithm with other methods such as off-line optimization, constant rate, constant quantization, and constant distortion using image set 2 composed of randomly chosen 30 images: (a), (b) PSNR of each image, (c), (d) bit rate of each image and (e), (f) memory usage for this image set.

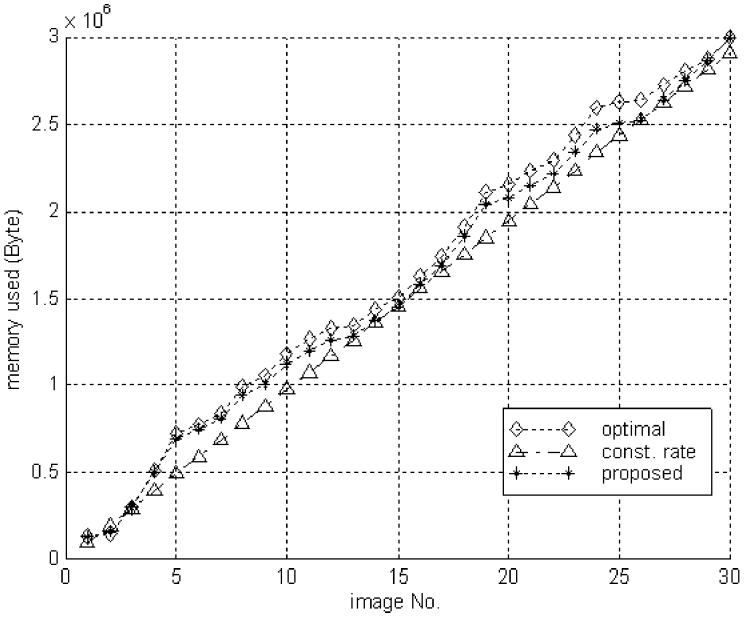


(c)

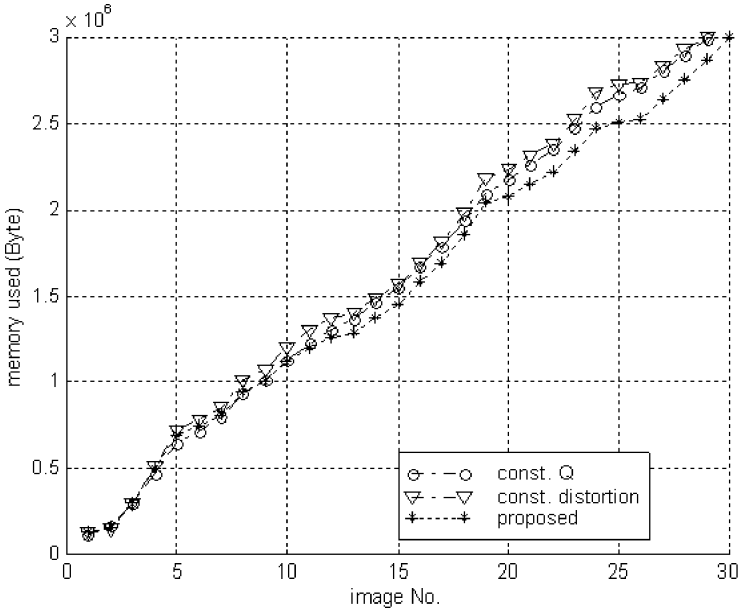


(d)

Figure 3.5 - continued

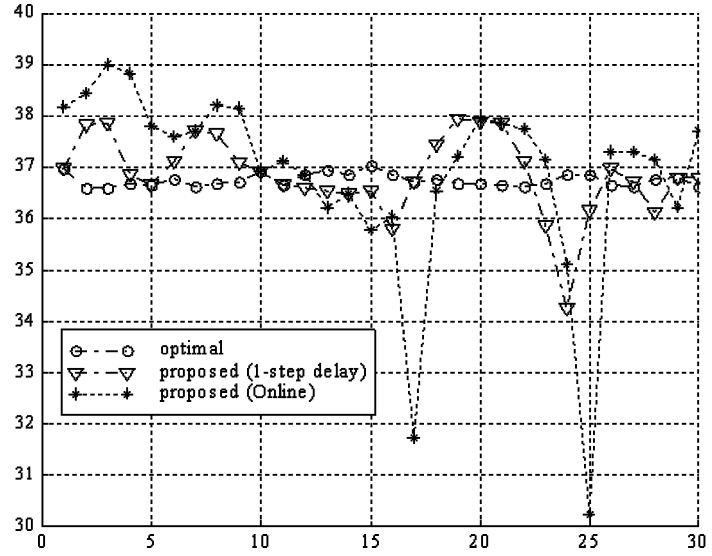


(e)

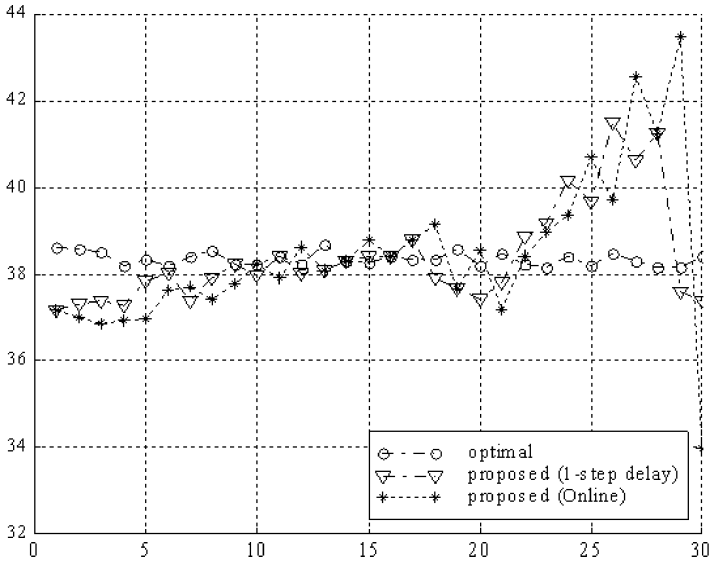


(f)

Figure 3.5 - continued

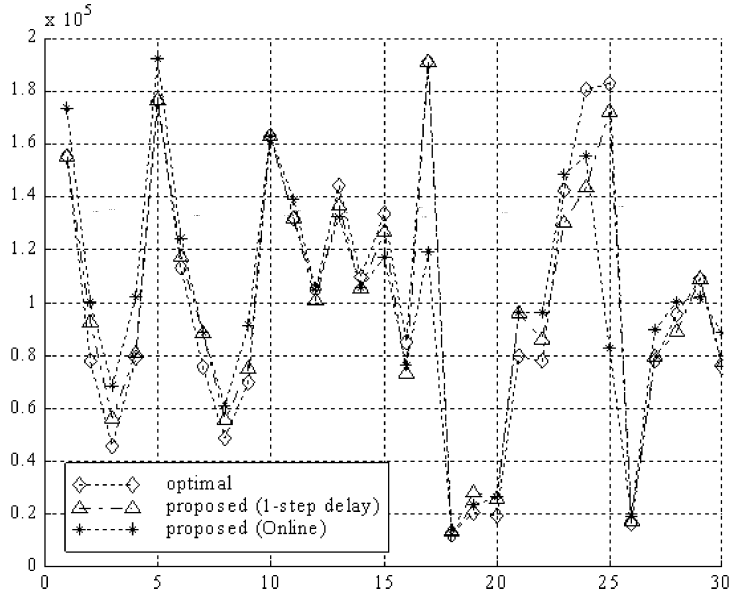


(a)

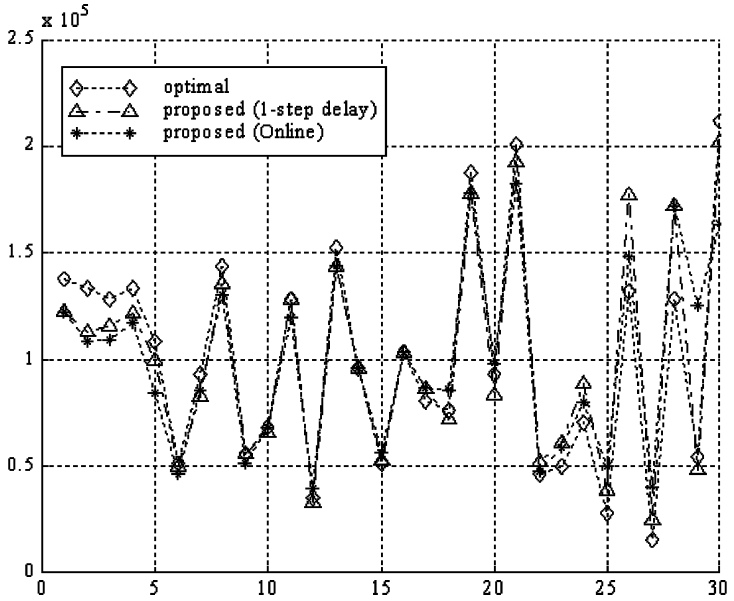


(b)

Figure 3.6: Performance comparison between 1 step delay and normal online methods with an off-line optimization method using two image sets: (a), (b) PSNR and (c), (d) bit rate of two different image sets. 1 step delay rate control uses the information of the next image for bit allocation of a current image.



(c)



(d)

Figure 3.6 - continued

Chapter 4

Optimal Rate Control for Video Transmission over CBR/VBR Channels based on a Hybrid MMAX/MMSE Criterion

In this chapter, we consider the problem of rate control for video transmission. We focus on finding off-line optimal rate control for constant bit-rate (CBR), and for variable bit-rate (VBR) transmission with a token bucket policing function. To ensure a maximum minimum quality is obtained over all data units, we use a minimization of maximum distortion (MMAX) criterion for this problem. We show that, due to the buffer and channel constraints, a MMAX solution leads to a relatively low average distortion, because the total rate budget is not completely utilized. Therefore, after finding a MMAX solution, an additional minimization of average distortion (MMAX+) criterion is proposed to increase overall quality of the data sequence by using the remaining resources, i.e. those resources that

were not utilized by the MMAX solution. The proposed algorithms lead to an increase in average quality with respect to the MMAX solution, while providing a much more constant quality than MMSE solutions. Moreover, we show how the MMAX+ approach can be implemented with low complexity.

4.1 Introduction

Future high bandwidth video applications, such as video-on-demand (VOD), will require transmission over the network of video compressed at a variable rate. Thus, a rate control has to be used, based on objectives such as coded video quality or data rate. Also, video transmission needs to be performed under delay constraints for real time playback, since video frames that arrive too late are useless.

To transmit VBR encoded data, VBR transmission is preferable to CBR transmission since VBR transmission needs lower end-to-end delay and a smaller buffer size [53, 5, 23]. However, due to the limited network resources, negotiation between each user and the network is indispensable in order to ensure QoS (quality of service) guarantees, where the parameters specified to define QoS can be delay jitter, bandwidth, end-to-end delay, and so on. In addition, policing mechanisms are used to alert the network if there are users who violate the agreed upon transmission parameters.

VBR video transmission through ATM (Asynchronous Transfer Mode) networks with a leaky bucket policing function has been studied in the literature [53, 46, 5, 23]. In [53, 5, 23], VBR transmission under both encoding and decoding buffer constraints and channel constraints is studied. In [46], multiple leaky bucket policing is introduced to regulate peak rate. It may be desirable to supplement a traffic policing function with a traffic shaping policy where traffic shaping is used to smooth out a traffic flow. One simple traffic shaping approach is "token bucket" (TB) policing. At the encoder side, leaky buckets and token buckets with the same parameters (i.e., the bucket size and the token rate) are equivalent, in that they impose the same constraints on the encoder. However, at the decoder side, the incoming data rate produced by each of these two approaches can be different.

Token buckets are also specified in next generation Internet Protocol (IP) networks. The Internet Engineering Task Force (IETF) has defined the Guaranteed Service (GS) in order to provide QoS to real time applications and token bucket policing is recommended as a traffic shaping method for GS [62].

After channel and source constraints, such as channel bandwidth, peak transmission rate, limited delay, total bit-budget or the size of codec buffers, have been determined, a target quality measure should be chosen. Most previous work for image and video coding has been based on minimization of average distortion (MMSE). As a consequence, optimal bit allocation under various constraints for

the MMSE criterion has been widely studied in the literature. Examples include bit allocation for arbitrary inputs and a discrete set of available quantizers [64], for dependent quantization [51], bit allocation under buffer constraints [48] and bit allocation for video transmission over ATM networks [5] [23]. A main drawback of the MMSE criterion is that the quality difference between frames can be large and some frames may be coded at relatively low quality even though the average quality over all frames is high. A minimization of maximum distortion (MMAX) criterion has been proposed to prevent this heavy fluctuation of source quality [58]. Solutions for the bit allocation problem under the MMAX criterion for both independent [41] and dependent quantizers [60] have been studied. Using this criterion, coding units having a significantly lower than average quality can be avoided. However, when multiple constraints are present, as when buffering is considered, the MMAX criterion by itself may be inefficient. This is because the MMAX optimization is terminated as soon as it cannot decrease the maximum overall distortion. For example, in the case of CBR transmission of a video sequence with buffer constraints, the maximum distortion frame could occur for a frame that is located in a period of several consecutive high complexity frames. Because these frames may require higher data rate than the given transmission rate, the buffer will tend to fill up. If this is the case the algorithm will stop because the distortion of the worst frame cannot be reduced without incurring

in overflow. This means that additional rate could be used in other parts of the sequence, so that overall quality could be increased.

A criterion for minimizing distortion in lexicographical sense (MLEX) has been proposed as a modified MMAX approach to increase overall quality [20] [19]. In a MLEX criterion, two different solutions are compared by arranging each solution as a sorted list of the achieved distortions in a non-increasing distortion order. Then a comparison of the distortions is based on considering the two lists starting from the 1^{st} index. If the distortions in the first position are equal then the 2^{nd} indices are compared. If the distortion of the two solutions is different for a given index then the solution with smaller distortion in that index is the better one. Otherwise the comparison is continued through the following position until the distortion of two solutions are different. This criterion is used to find optimal bit allocation under CBR constraints in [20] where quantizer levels are used as a distortion measure. Since all frames have the same set of quantizer levels, it is shown that the optimal solution is the one having constant quantization. It is also shown that under buffering constraints the optimal solution consists of segments with consecutive frames being allocated the same quantizer. In general, if a distortion measure can take any arbitrary values, the proposed algorithm cannot be easily applied. In our work, we show how the algorithm used to find the MMAX solution with buffer constraints can be extended to find the MLEX solution as well.

As an alternative approach to increase overall quality after finding a MMAX solution, we propose to use a MMSE criterion for the remaining bit-budget. We denote this criterion MMAX+, because it adds additional targets to the MMAX criterion. Note that in [58] a MMSE criterion is used to break the tie among several MMAX solutions in a bit-budget constrained problem. However, in that work there is no additional bit-budget to be reallocated.

Both MMAX+ and MLEX will increase the average quality with respect to the MMAX solution (assuming an additional bit-budget is available.) However, since MMAX+ explicitly targets average distortion it will lead to better average MSE than MLEX.

In this chapter, extending our previous work in [36, 37], we propose an offline optimal rate control algorithm in MMAX and MMAX+ criteria for video transmission over CBR and VBR channel with a discrete set of quantizers available to code each frame. We introduce MMAX and MMAX+ criteria in these buffer-constrained (for CBR transmission) and channel-constrained (for VBR transmission) problems, so that the best minimum quality of all frames is provided by the MMAX criterion and good overall quality is achieved by the MMAX+ criterion. We also propose an algorithm to reduce the complexity of finding the MMAX+ solution. Simulation results show that the solution of our proposed method gives almost the same average quality as the MMSE solution and much better minimum quality, with lower complexity.

This chapter is organized as follows: in Section 2, algorithms to find the optimal solution for CBR transmission are proposed. In Section 3, algorithms to find the optimal solution for VBR transmission with a token bucket policing function are proposed. Experimental results are provided in Section 4. Conclusions are provided in Section 5.

4.2 Rate Control for video transmission over CBR Channels

4.2.1 Optimal rate control for a MMAX criterion

Video transmission is constrained by the maximum delay allowable, the encoder and decoder buffers and channel constraints such as channel rate and channel policing functions. In the CBR transmission case, it is possible to prevent the decoder buffer from underflowing (or overflowing) by preventing the encoder buffer from underflowing (or overflowing) [53]. Therefore, the constraints of this problem are the transmission rate (C) and the encoder buffer size (B) since the delay is determined by C and B in a CBR case. We assume that one frame is coded every T seconds and immediately moved to the encoder buffer after encoding. Then

the problem we are trying to solve using a MMAX criterion can be formulated as

$$\min_{q_i}(\max(D_i)) \text{ s.t. } B_i \leq B \text{ for all } i, \tag{4.1}$$

where D_i is the distortion of the i^{th} frame ($1 \leq i \leq S$, S is the number of frames), B_i is the buffer occupancy after the encoded i^{th} frame is moved to the buffer (i.e., $B_i = \max(B_{i-1} + R_i - C \cdot T,\ 0)$, where R_i is the bit-budget of the i^{th} frame.) For a given frame i, the $(R_i(q_i),\ D_i(q_i))$ pairs are determined by the selection of a quantization level j ($1 \leq j \leq Q_i$). The algorithm to find the optimal MMAX solution can then be defined as follows:

Algorithm 1: *Optimal bit allocation in a CBR channel with buffer constraints under a MMAX criterion*

[Step 0]: *Initialize the buffer occupancy by quantizing all frames with the coarsest quantization available to each frame.*

[Step 1]:*Find the frame that has maximum distortion and decrease the quantization step size of that frame.*

[Step 2]:*If the buffer is not in overflow then go to Step 1, otherwise STOP. The frame that has maximum distortion is the frame whose quantization changed just before buffer overflow. Obviously, the maximum distortion is the distortion of that frame without the final quantization change.*

The bisection algorithm [58] can be also applied to this problem. Note, however, that this algorithm may not be terminated since it is based on the bisection of distortions, and distortions can take arbitrary positive real values. This algorithm can be modified by pre-sorting all possible R-D data of all frames by a non-increasing distortion order. After sorting the data, a bisection method is applied to the sorted indices in order to find the optimal solution. This modified algorithm can be described as follows:

Algorithm 2: *Optimal bit-allocation in a CBR channel with buffer constraints under a MMAX criterion: Pre-sorting and Bisection*

[Step 0]: *Sort all R-D data of all frames in a non-increasing distortion order, where the data in the sorted array are rates, distortions, quantization and frame numbers. Initialize buffer occupancy by quantizing all frames with the coarsest quantization available to each frame. Set this choice as a solution. Set the largest index of chosen quantization of all frames in the sorted array as* Min*. Set* Mid *to* $\lceil (Min + Max)/2 \rceil$*, where* Max *is the maximum index of the array.*

[Step 1]:*For each frame, choose the quantizer that has the lowest distortion among all the quantizers for that frame having an index lower than or equal to* Mid*.*

[Step 2]:*If the buffer is not in overflow then update the solution and let* $Min =$

Mid and $Mid = \lceil (Mid + Max)/2 \rceil$ *else let* $Max = Mid$ *and* $Mid = \lceil (Min + Mid)/2 \rceil$. *If Mid is not changed then STOP, otherwise go to Step 1.*

Under the assumption that rate and distortion of all quantization levels are pre-calculated, in Algorithm 1 Step 0 needs S selections, Step 1 needs $\log S$ comparisons to find the frame that has maximum distortion and S comparisons are needed to check buffer overflow in Step 2. Since the number of iterations is at most SQ where Q is the maximum of Q_i, the complexity of Algorithm 1 is $O(S^2Q)$. But since the algorithm is terminated when it cannot improve the maximum distortion, in general, the complexity is much lower than this bound.

In Algorithm 2, the complexity of merge-sorting is $SQ \log S$ since the data of each frame are already sorted (see Fig. 1.7). In Step 1, complexity of the entire iteration is at most SQ since in each iteration already checked data do not need to be checked again. The number of iterations is at most $\log SQ$ and Step 2 needs $O(S)$ comparisons in each iteration, so that total complexity of Step 2 is $O(S \log SQ)$. Therefore the complexity of Algorithm 2 is $O(SQ \log S)$ whereas the complexity of the MMSE algorithm of this problem is $O(BSQ)$ [48]. Since in a video application, B is relatively large, the complexity of the MMAX algorithm can be much lower than that of the MMSE algorithm.

The optimal MMAX solution may result in buffer underflow, especially in the case when several easily compressed frames are coded successively. Because underflow occurs at the encoder, we can use stuffing bits to prevent any problem. In this chapter, we propose to use this "spare" bit-budget due to underflow in order to decrease the mean square error (MSE). We term this the MMAX+ approach as the MMAX solution is improved upon with an additional MSE criterion. MSE's limitations are well known but, obviously, our MMAX+ technique could also be used with alternative additive distortion metrics.

4.2.2 Optimal rate control for a MMAX+ criterion

After finding the MMAX solution, the problem we are trying to solve using a MMAX+ criterion can be formulated as

$$\min_{q_i}(\sum_{i=1}^{S} D_i) \text{ s.t. } B_i^M \leq B_i \leq B \text{ for all } i, \tag{4.2}$$

where B_i^M is the buffer state at the i^{th} frame time when the MMAX solution is used. Note that we take the MMAX solution as the initial condition and we never reduce the bit allocation to a frame chosen by the MMAX approach, i.e., the additional step we propose can only increase the number of bits used for each frame.

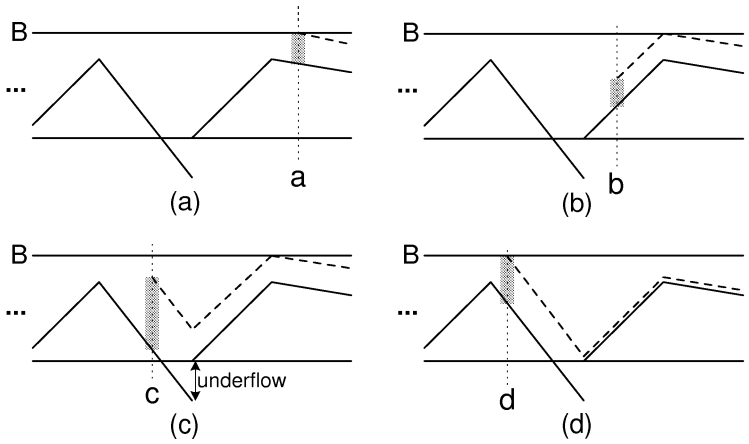


Figure 4.1: Examples of computation of the effective buffer size (EBS) of different frames. The solid line represents the buffer occupancy of a MMAX solution. The height of the gray box is the EBS of the given frame and dashed lines show that the determined EBS does not induce buffer overflow. The EBS of frame "a" is determined by the residual buffer of the frame and that of frame "b" is determined by the residual buffer of a following frame. For frames "a" and "b", the EBS is determined by the minimum residual buffer size of the current and following frames. The EBS of frame "c" is determined by the sum of the amount of underflow and the EBS of a frame after underflow. The EBS of frame "d" is determined by the residual buffer size of the frame because it is smaller than the EBS of the following frame.

It is important to note that the upper bound constraint in (4.2) is not tight. This is because the trace B_i^M already incorporates the effect of transmitted bits. Thus any increase to B_i over B_i^M leads (if there was no buffer underflow) to an increase in $B_{i'}$ for $i' > i$, so that the overflow constraint could be violated for i', even if it is not for i (see Fig. 4.1 (b)). To reduce the upper bound of the buffer state of a frame, we introduce the concept of effective buffer size (EBS), where the EBS of a frame is the maximum bit-budget that can be used to increase the quality of the frame and such that no overflow occurs. Obviously EBS_i (the EBS

of the i^{th} frame) is smaller than or equal to the residual buffer size after selecting a quantizer according to the MMAX solution of the frame (RBS_i), the difference between the maximum buffer size and the buffer state of the MMAX solution (i.e., $RBS_i = B - B_i^M$), which varies from frame to frame. RBS_i can also be explained as the maximum amount of bits we can use for the i^{th} frame without leading to an end-to-end delay violation. Examples of computation of the EBS are shown in Fig. 4.1. In the figure, the EBS of frame "b" is determined by the minimum RBS of all frames from "b" onwards. This is because additional bits used at "b" will increase the buffer occupancy of all following frames (dashed lines in Fig. 4.1 (b)). However, if the buffer is in underflow at the i^{th} frame interval (where the i^{th} frame interval means the interval between the i^{th} frame and the next frame) then the amount of underflow (UF_i, where $UF_i = \max(C \cdot T - B_i, 0)$) can also be added to the bit-budget of the i^{th} frame without affecting the buffer state of future frames (see Fig. 4.1 (c)). In this example, given that EBS_{i+1} is known, EBS_i is obtained as $EBS_i = \min(RBS_i, UF_i + EBS_{i+1})$. The EBS for a frame can be formed based on the following theorem.

Theorem 1: The EBS for a frame can be formed as

$$EBS_i = \begin{cases} RBS_S & : i \text{ is the last frame,} \\ \min(RBS_i, UF_i + EBS_{i+1}) & : \text{otherwise.} \end{cases} \tag{4.3}$$

Proof) At first, we consider the last frame. Since the buffer state is causal, increasing the rate for a frame only affects the buffer state of the current and future frames. Since no future frame exists, the EBS of the current frame is restricted by the buffer overflow of the current frame and it is determined by the RBS of the current frame (i.e., $EBS_S = RBS_S$.) Next, we use induction to prove the result. Since increasing the rate for a frame does not affect the buffer state of the previous frames, the solution is to choose the maximum bit-budget that does not result in buffer overflow at the current and future frames. For any i $(1 \leq i \leq S-1)$, assume EBS_{i+1} is known. Then EBS_{i+1} guarantees no buffer overflow in all future frames. If the buffer is in underflow at the i^{th} frame interval then the rate can be increased by the amount of underflow without changing the buffer state of the future frames. So in order to consider the overflow of future frames only, the solution is the sum of the amount of underflow at the i^{th} frame

interval (UF_i) and EBS_{i+1}. Since the rate can be increased at most the remaining buffer size of the i^{th} frame (RBS_i), EBS_i is determined by (4.3) ■

Therefore, the EBS is computed from the last frame by using the equation (4.3). After computing the EBS for all frames, the optimization problem following the MMAX+ criterion is redefined as

$$\min_{q_i}\Big(\sum_{i=1}^{S} D_i\Big) \text{ s.t. } B_i^M \le B_i \le EBS_i + B_i^M \text{ for all } i. \tag{4.4}$$

This new formulation now guarantees that increasing B_i does not lead to overflow. The allowable quantization levels of the i^{th} frame (q_i) are also reduced to ($q_i^M \le q_i \le Q_i'$), where q_i^M and Q_i' are determined by the MMAX solution and the upper bound of B_i, respectively.

This rate control problem can be solved by using a dynamic programming method [48] or a Lagrangian optimization method [45, 5]. Given the buffer constraints due to our goal to preserve the MMAX solution, the number of states in a dynamic programming method can be reduced significantly by computing the EBS. As mentioned in the previous section, the complexity of the MMSE algorithm is proportional to B and Q. Since the EBS and the number of allowable quantization levels are much smaller than B and Q, the complexity of the MMAX+ algorithm is much lower than that of the MMSE algorithm.

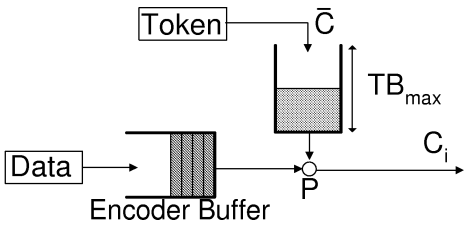


Figure 4.2: System model of TB policing. $\bar{C}$ is the token rate and C_i is the transmission rate of the i^{th} frame interval. TB_{max} and P indicate the size of a token bucket and the peak rate respectively. In this policing, one byte data can be transmitted per token.

4.3 Rate Control for video transmission over VBR Channels

4.3.1 Optimal rate control in a MMAX criterion

In a VBR transmission case, preventing encoder buffer overflow does not guarantee that a decoder buffer is not in overflow or underflow. Therefore, the decoder buffer state has to be considered as a constraint in a rate control scheme for VBR transmission [5] [23]. Here, to maximize channel utilization, we assume that transmission is constrained by the maximum delay, rather than by the size of encoder and decoder buffers. In other words, the size of encoder and decoder buffers is assumed to be large enough to always store all the data that it will be possible to transmit under the given delay constraint (for the given channel constraints). Thus, our goal is finding the optimal rate control for the MMAX and MMAX+ criteria under the given maximum delay and network policing constraints.

In this chapter, TB policing is used as a policing constraint. TB policing is defined with 5 parameters named transmission specification (Tspec) in [62], [63]. In the specification, the amount of data sent is constrained not only by the available tokens but also by the peak rate. This peak rate constraint is used to put a limit on the size of data bursts. Therefore, as shown in Fig. 4.2, the constraints of our problem are token bucket parameters (the token rate ($\bar{C}$) and the size of token bucket (TB_{max})), the peak rate ($P,\ P \geq \bar{C}$) and the maximum delay (M), where the peak rate and the token rate are measured in bytes per frame interval and the maximum delay is measured in frame intervals. Then the problem we are trying to solve using a MMAX criterion can be formulated as follows:

$$\min_{q_i}(\max(D_i)) \tag{4.5}$$

$$\text{s.t. } B_i \leq \min(TB_i + M \cdot \bar{C},\ M \cdot P), \tag{4.6}$$

$$\text{where } TB_i = \min(TB_{max}, TB_{i-1} + \bar{C} - C_{i-1}), \tag{4.7}$$

$$B_i = B_{i-1} + R_i - C_{i-1}, \tag{4.8}$$

$$\text{with } C_{i-1} = \min(B_{i-1},\ TB_{i-1} + \bar{C},\ P). \tag{4.9}$$

In the above equations, B_i indicates the buffer occupancy after the encoded i^{th} frame ($1 \leq i \leq S$, S is the number of frames) is moved to the buffer, TB_i indicates the TB state just before starting the i^{th} frame interval (the interval between the

i^{th} frame and the next frame) transmission, and M indicates maximum delay in frame units. Initial transmission rate (C_0) is zero and initial encoder buffer and TB state can be any values between zero and their maximum value. The number of tokens in a TB (TB_i) and the number of bits in an encoder buffer (B_i) cannot be negative; this is guaranteed by (4.9). Among many possible channel rate selection policies, in (4.9), we select the maximum available transmission rate at each frame interval. This selection guarantees a performance as good as the best, since it tends to minimize the token overflow probability (because it uses a transmission rate that is as high as possible.) Instead of using the transmission rate in (4.9), if we select a reduced transmission rate for certain frame intervals then both the token count in the TB and the number of bits in the encoder buffer increase. Since these bits are generated by the optimal solution they should be sent, otherwise, the optimal solution cannot be achieved. Thus, the extra tokens in TB will be still needed later on to transmit the additional bits in the buffer. Therefore, by using different transmission policies, we cannot get any extra channel capacity but will have higher probability of TB overflow and delay violation. Note, however, that the optimal solution is guaranteed by the selection in (4.9) only in the case that there are no additional constraints, such as the size of an encoder or decoder buffer, or maintaining a near constant transmission rate [11].

Also, in order to prevent decoder buffer underflow, B_i should be lower than or equal to $\sum_{k=i}^{i+M-1} C_k$ [23]. The following Lemma shows that the channel rate selection policy used never produce decoder buffer underflow.

Lemma 1: For any $\{R_i \text{ s.t. } B_i \leq \min(TB_i + M \cdot \bar{C}, M \cdot P),\ 1 \leq i \leq S\}$ the channel rate selection policy in (4.9) never results in decoder buffer underflow.

Proof) As mentioned above, in order to guarantee no decoder buffer underflow, B_i should satisfy the following condition.

$$B_i \leq \sum_{k=i}^{i+M-1} C_k \ . \tag{4.10}$$

Based on (4.9), C_i can be one out of three possible choices, so the number of outcomes of the right side in (4.10) is 3^M and we need to show that each of these outcomes can satisfy the above condition or that, otherwise, it cannot be a possible outcome.

i) Let us assume that C_j is B_j, and for all m such that $i \leq m < j$, C_m is $TB_m + \bar{C}$ or P, where j can be $i \leq j \leq i + M - 1$. Then from (4.8), C_j can be written as

$$\begin{aligned} C_j &= B_j = B_{j-1} + R_j - C_{j-1} \\ &= B_i + \textstyle\sum_{l=i+1}^{j} R_l - \sum_{l=i}^{j-1} C_l \ , \end{aligned} \tag{4.11}$$

and the sum of channel rates can be

$$\begin{aligned}\sum_{k=i}^{i+M-1} C_k &= \sum_{m=i}^{j-1} C_m + (B_i + \sum_{l=i+1}^{j} R_l - \sum_{l=i}^{j-1} C_l) + \sum_{l=j+1}^{i+M-1} C_l \\ &= B_i + \sum_{l=i+1}^{j} R_l + \sum_{l=j+1}^{i+M-1} C_l \geq B_i\end{aligned} \tag{4.12}$$

Therefore, at any time j, if the channel rate is chosen as the first term of (4.9) (i.e., B_j) then the condition is always satisfied regardless of the channel rate at time l, $j < l < i + M - 1$. This can be explained because this choice, $C_j = B_j$, results in all the data in the buffer being fully transmitted.

ii) Let us assume that the channel rate chosen will always be the second or third choice in (4.9), i.e., there is no j such that $C_j = B_j$. Also we assume that for some j $C_j = TB_j + \bar{C}$, and for all m such that $i \leq m < j$, $C_m = P$, where j can be $i \leq j \leq i + M - 1$. This means that all the tokens in TB are used at the j^{th} frame time interval. Therefore C_{j+1} is $\bar{C}$ (since $TB_{j+1} = 0$) or P. Since P is larger than or equal to $\bar{C}$, C_{j+1} should be $\bar{C}$. Similarly C_l for all l such that $j + 1 \leq l \leq i + M - 1$ has to be $\bar{C}$. Therefore once the second term is chosen then all channel rates after this are fixed as $\bar{C}$. In other words, among all 3^M combinations, those combinations choosing the third term after the second term is chosen are not possible.

By using (4.7), C_j can be rewritten as

$$C_j = TB_j + \bar{C} = \min(TB_{max}, TB_{j-1} + \bar{C} - C_{j-1}) + \bar{C} \ . \tag{4.13}$$

Based on our assumptions, $C_{j-1} = P$, and therefore $C_j = TB_{j-1} + 2\bar{C} - P$ (since P is larger than or equal to $\bar{C}$ and TB_{j-1} is at most TB_{max}.) By continuously substituting the TB state, C_j can be found to be :

$$C_j = TB_i + (j - i + 1) \cdot \bar{C} - (j - i) \cdot P \tag{4.14}$$

and the sum of the channel rates is

$$\begin{aligned} \sum_{k=i}^{i+M-1} C_k &= (j-i) \cdot P + \{TB_i + (j-i+1) \cdot \bar{C} - (j-i) \cdot P\} + (i+M-1-j) \cdot \bar{C} \\ &= TB_i + M \cdot \bar{C} \geq B_i \ , \end{aligned} \tag{4.15}$$

with the last inequality being a consequence of (4.6).

iii) The only possible outcome that we have not considered so far is that where all the channel rates are P. In this case, the sum of the channel rates is $M \cdot P$ and the condition is satisfied given (4.6).

From (i), (ii) and (iii), we proved that any possible combination of the channel rate never induce the decoder buffer underflow, where the channel rate allocation of (4.9) is used. Therefore we concentrate on the delay constraints and assume that the channel rate is chosen following (4.9) ■

As in (4.6), the encoder buffer state is constrained by the maximum amount of data that can be sent during the next M frame intervals. Therefore the minimum

size of the encoder buffer (B) that will prevent additional buffer constraints from arising is

$$B = \min(TB_{max} + M \cdot \bar{C},\ M \cdot P). \tag{4.16}$$

In other words if the physical buffer size is greater than B in (4.16) then we need not consider the additional buffer constraint (but still consider the delay constraint as in (4.6)). The algorithm to find the optimal MMAX solution can then be defined as follows:

Algorithm 3: *Optimal bit allocation in a VBR channel with channel constraints under a MMAX criterion*

[Step 0]: *Initialize the buffer occupancy (B_i) by quantizing all frames with the coarsest quantization available to each frame.*

[Step 1]: *Find the frame that has maximum distortion and decrease the quantization step size of that frame.*

[Step 2]: *If the buffer state satisfies the condition in (4.6) for all i then go to Step 1, otherwise STOP. The frame that has maximum distortion is the frame whose quantization changed just before STOP.*

Fig. 4.3 shows two simple examples of VBR transmission with TB policing in a MMAX criterion. As shown in (a), even though more data are stored in the

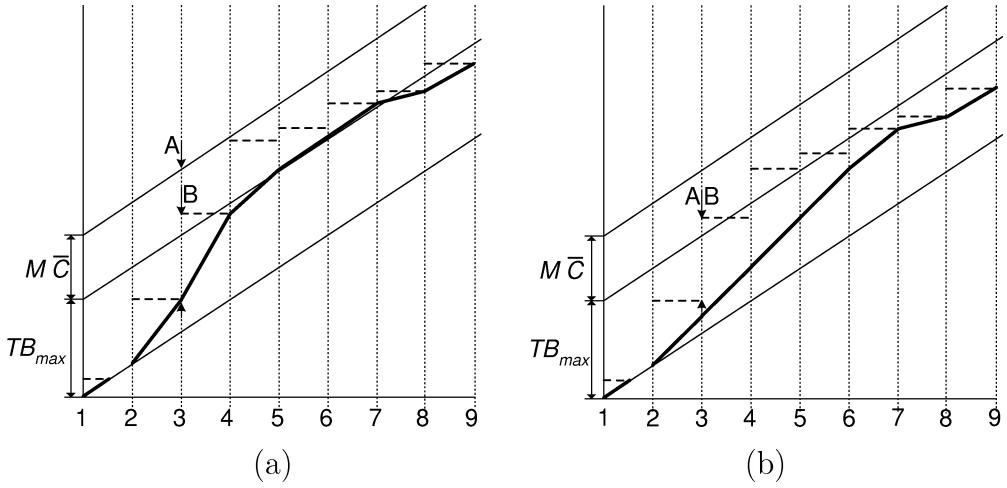


Figure 4.3: VBR transmission with TB policing with parameters ($\bar{C}$, TB_{max}, P) under a MMAX criterion. Horizontal axes indicate time in frame units and vertical axes indicate the size of transmitted data. Horizontal dashed lines indicate the solutions in a MMAX criterion and the slopes of thick lines indicate transmission rate of each frame interval. (a) is the case that the peak rate is high enough not to be a constraint (i.e., $P \geq TB_{max} + \bar{C}$) and (b) is the case that the peak rate is used as a constraint.

buffer, the amount of data transmitted is always lower than the middle diagonal lines in Fig. 4.3 since the available maximum rate in a frame interval is determined by the remaining tokens in the TB and the new incoming tokens in the frame interval (i.e., $TB_i + \bar{C}$.) In (b), the constant slope between the 2^{nd} frame and the 6^{th} frame indicates the peak rate. As shown in the figure, the amount of transmitted data is limited even though more tokens are available.

The MMAX solution may result in TB overflow, especially when easily compressed frames are coded successively. Note that tokens that are dropped due to TB overflow cannot be used for future data transmission. Similar to the CBR

transmission in which the bit-budget due to underflow is used to decrease MSE, we use these "spare" tokens, which are not used due to token bucket overflow, in order to decrease MSE.

4.3.2 Optimal rate control in a MMAX+ criterion

Similar to the CBR case, after finding the MMAX solution, the problem can be formulated as

$$\min_{q_i}(\sum_{i=1}^{S} D_i) \text{ s.t. } B_i^M \leq B_i \leq B_i^U \text{ for all } i, \tag{4.17}$$

where B_i^M is the buffer state of the MMAX solution at the i^{th} frame time of the MMAX solution and B_i^U represents the right side of (4.6). Fig. 4.3 shows examples of B_i^U. In the figure, "A" and "B" indicate B_3^U and B_3^M respectively (in (b), "A" and "B" are same).

Note that although B_i^U in (4.17) is lower than or equal to B, this does not mean we can allocate additional $B_i^U - B_i^M$ bytes. For instance, if we take B_i^U as B_i then after transmitting data during the i^{th} frame interval (i.e., subtracting C_i) and storing the $(i+1)^{th}$ frame data (i.e., adding R_{i+1}^M), B_{i+1} may be larger than B_{i+1}^U and the delay constraint will be violated. This is because the trace B_i^M already incorporates the effect of transmitted data, and thus additional data allocated by the encoder does not result in additional channel rate (i.e., in order to increase the rate we need to exploit instances of token buffer overflow when

available transmission capacity was wasted.) Thus any increase to B_i over B_i^M leads to decreasing tokens in the TB or increasing data in the encoder buffer for all $i' > i$, so that delay violation could occur for i', even if it does not occur for i.

To reduce the upper bound of the buffer state of a frame, we introduce the EBS similar to the CBR case, where the EBS of a frame is the maximum additional rate that can be used to increase the quality of the frame and such that no violation of the condition in (4.6) occurs. In the VBR case, finding the EBS is more complicated since we need to consider the TB and encoder buffer states together instead of considering the encoder buffer state only as in the CBR case. In addition, the peak rate constraint needs to be considered.

We introduce a method named assigning latest coming tokens first (ALTF), in which to find EBS_i (the EBS of the i^{th} frame), R_j^M for all $j > i$ is transmitted by using the latest coming tokens first and R_k^M for all $k \leq i$ is transmitted by using the channel rate selection policy in (4.9). The main idea of the ALTF is saving as many tokens as possible that can be used for transmitting the data of the given frame data. Note that only the tokens in TB at the i^{th} frame time and the tokens coming between the i^{th} and $(i+M)^{th}$ frame time can be used for transmitting the i^{th} frame data. Fig. 4.4 shows an example of finding EBS_i by using ALTF. In the figure, the top and middle figures show the TB state and the arrival time of tokens, respectively, after applying the MMAX solution with the channel rate selection policy in (4.9). After the $(i+1)^{th}$ frame interval, TB

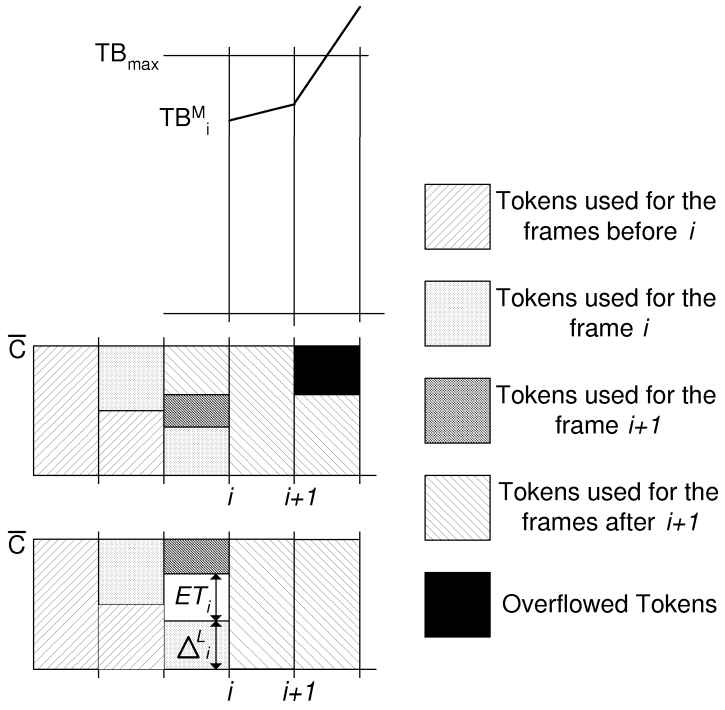


Figure 4.4: The top figure shows the TB state of the i^{th} and $(i+1)^{th}$ frame intervals. The middle and bottom figures show the arrival time of the tokens which are used to transmit i^{th} and $(i+1)^{th}$ frame data with the channel rate selection police in (4.9) and the ALTF method to find EBS_i, respectively. The vertical axis indicates the tokens coming in each frame interval and the tokens at the lower part of a frame interval arrive earlier than the tokens at the upper part of the frame interval.

(corresponding to the height of the black area) is in overflow and some tokens cannot be used for transmitting the data coming after overflow. After applying ALTF to find EBS_i, as in the bottom figure in Fig. 4.4, we can find the amount of extra tokens (ET_i) that we can use for transmitting additional i^{th} frame data without leading to any violation of transmission constraints for data coming after i^{th} frame. As shown in the figure, the tokens arriving late are used first in order to transmit the $(i+1)^{th}$ and following frame data.

But in order to calculate ET_i, we need to know the arrival time of the last token used for transmitting R_i^M and that of the first token used for transmitting R_{i+1}^M after applying ALTF, since ET_i can be determined as the number of tokens arriving between the two tokens.

The arrival time of the last token used for transmitting R_i^M can be determined by using TB_i^M and B_i^M. Since TB_i^M and B_i^M are not related to the data coming after the i^{th} frame, after applying ALTF to find the EBS of the i^{th} frame, TB_i^M and B_i^M are not changed. Note that ALTF for the i^{th} frame changes the transmission policy for the $(i+1)^{th}$ and the following frames only. The frame interval in which the token arrives (FI_i^L) and the location of the token in the interval ($\triangle_i^L$) can be determined as

$$\begin{aligned} FI_i^L &= i + \lfloor \tfrac{B_i^M - TB_i^M}{\bar{C}} \rfloor \ , \\ \triangle_i^L &= (B_i^M - TB_i^M) \% \bar{C} \ , \end{aligned} \tag{4.18}$$

where a % operator indicates the remainder after division. The equation shows that if the amount of data in the buffer is larger than the number of tokens in the TB then the tokens coming after the i^{th} frame time should be assigned for the data in the buffer. Therefore in this case, FI_i^L should be larger than or equal to i. In Fig. 4.5, FI_i^L is $i-1$ and this means, at the i^{th} frame time, the number of tokens in TB is larger than the number of data in the buffer.

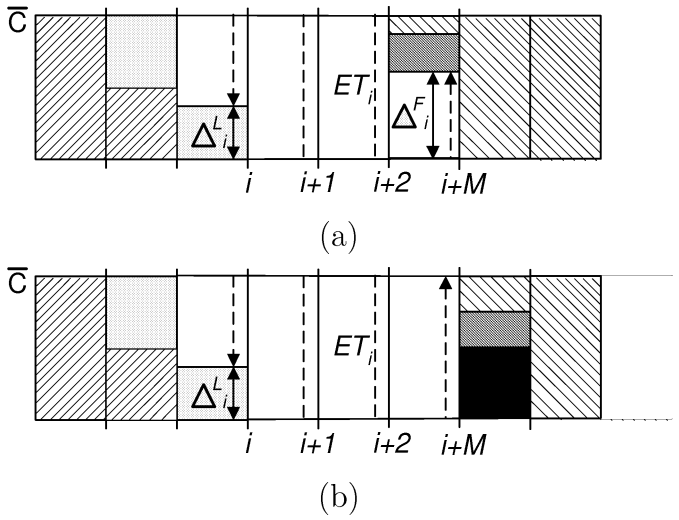


Figure 4.5: Illustrations of $\triangle_i^L$, FI_i^L, $\triangle_i^F$, FI_i^F and ET_i. In (b), tokens corresponding to the black area cannot be used for transmitting the i^{th} frame data due to the delay constraint.

Next, in order to find the arrival time of the first token used for transmitting R_{i+1}^M, we need to assign tokens to R_j^M for all $j > i$ starting from the last frame data (R_S^M). We define the frame interval in which the first token for R_{i+1}^M arrives as FI_i^F and the location of the token in the interval as $\triangle_i^F$. The FI_i^F and $\triangle_i^F$ are determined from FI_{i+1}^F and $\triangle_{i+1}^F$ by assigning tokens to R_{i+1}^M as follows.

$$\begin{aligned} FI_i^F &= FI_{i+1}^F + \lfloor \frac{\triangle_{i+1}^F - R_{i+1}^M}{\bar{C}} \rfloor \ , \\ \triangle_i^F &= (\triangle_{i+1}^F - R_{i+1}^M) \% \bar{C} \ . \end{aligned} \tag{4.19}$$

For example, in Fig. 4.5 (a), FI_{i+1}^F and FI_i^F are $i+2$. In (4.19), since $\triangle_{i+1}^F$ is always smaller than $\bar{C}$, the difference between $\triangle_{i+1}^F$ and R_{i+1}^M is smaller than $\bar{C}$

and FI_i^F cannot be larger than FI_{i+1}^F. This is obvious because we first assign tokens coming later. But we cannot use the tokens coming after the limited delay (M), in other words, some tokens cannot be used for transmitting the i^{th} frame data if FI_i^F is larger than or equal to $i + M$. Therefore in this case, we change FI_i^F and $\triangle_i^F$ to $i+M$ and 0 respectively. In Fig. 4.5 (b), the tokens corresponding to the black area cannot be used due to the delay constraint and $\triangle_i^F$ is changed to 0.

Since FI_i^F and $\triangle_i^F$ are calculated recursively, we need to know the initial values (i.e., FI_S^F and $\triangle_S^F$). These values depends on the final encoder buffer (B_f) and TB states (TB_f) at the $(S+1)^{th}$ frame time, which are given as constraints. If TB_f is identical to B_f (i.e., the amount of tokens in TB is exactly same as that of data in the buffer) then remaining tokens are fully used for transmitting the remaining data. This means that tokens do not need to be stored in the TB for transmitting the future data. But if TB_f is larger than B_f then this constraint means that the number of tokens corresponding to the difference between TB_f and B_f should be stored for future use. In this case, we can consider that the same number of tokens are used for transmitting the $(S+1)^{th}$ frame data (although it does not exist). As an opposite case, if B_f is larger than TB_f then we can consider that the amount of future tokens corresponding to the difference between B_f and

TB_f are available after assigning tokens to the $(S+1)^{th}$ frame data. Therefore similar to finding FI_i^L and $\triangle_i^L$, FI_S^F and $\triangle_S^F$ are determined as

$$\begin{aligned} FI_S^F &= S + 1 + \lfloor \tfrac{B_f - TB_f}{\bar{C}} \rfloor \ , \\ \triangle_S^F &= (B_f - TB_f) \% \bar{C} \ . \end{aligned} \tag{4.20}$$

Again, if $FI_S^F \geq S + M$ then

$$\begin{aligned} FI_S^F &= S + M \ , \\ \triangle_S^F &= 0 \ . \end{aligned} \tag{4.21}$$

After finding the arrival time of the two tokens, ET_i is determined as

$$ET_i = (FI_i^F - FI_i^L) \cdot \bar{C} + \triangle_i^F - \triangle_i^L \ . \tag{4.22}$$

EBS_i is constrained not only by the maximum number of available tokens that does not induce any violation (i.e., ET_i) but also by the maximum available channel bandwidth, which is limited by the peak rate. Therefore EBS_i is determined as

$$EBS_i = \min(ET_i, M \cdot P - B_i^M) \ . \tag{4.23}$$

After computing the EBS for all frames, the problem in a MMAX+ criterion is redefined as

$$\min_{q_i}(\sum_{i=1}^{S} D_i) \text{ s.t. } B_i^M \leq B_i \leq EBS_i + B_i^M \text{ for all } i, \tag{4.24}$$

This new formulation now guarantees that increasing B_i does not lead to TB underflow and any violation of the condition in (4.6).

Similar to the encoder buffer state, we can find the lower and upper bounds of the TB state. As we mentioned, we cannot save more tokens than TB_i^M without increasing data in the encoder buffer (i.e., without changing the channel rate selection policy for the i^{th} and previous frames.) Therefore TB_i^M is the upper bound of TB_i. The lower bound of the TB state can be determined by using FI^F and $\triangle^F$. Since TB_i indicates the number of tokens in TB before starting the i^{th} frame interval, TB_i should be large enough not to result in constraints violations when make transmitting the MMAX solution of the i^{th} and following frames. If FI_{i-1}^F is larger than or equal to i then the data of current and future frames can be sent without using the tokens coming before the i^{th} frame time (note that FI_{i-1}^F relates to the tokens used for the i^{th} and following frames.) In other words, TB_i can be zero in case that the buffer is empty before adding i^{th} frame data. But if $FI_{i-1}^F < i$ then some tokens should be stored in TB for future use, otherwise the MMAX solution cannot be preserved. The minimum use of

tokens in TB for the i^{th} and following frames is determined by using FI^F_{i-1} and $\triangle^F_{i-1}$, since we first assign tokens arriving later from the last frame in order to find FI^F and $\triangle^F$. Therefore the lower bound of the TB state (TB^L) can be determined as

$$TB^L_i = \begin{cases} (i - FI^F_{i-1}) \cdot \bar{C} - \triangle^F_{i-1} & , \text{ if } FI^F_{i-1} < i, \\ 0 & , \text{ otherwise,} \end{cases} \tag{4.25}$$

where TB^L_i is the TB^L of the i^{th} frame time. As a result, the TB state of the i^{th} frame time is bounded as

$$TB^L_i \leq TB_i \leq TB^M_i \ , \text{ for all } i. \tag{4.26}$$

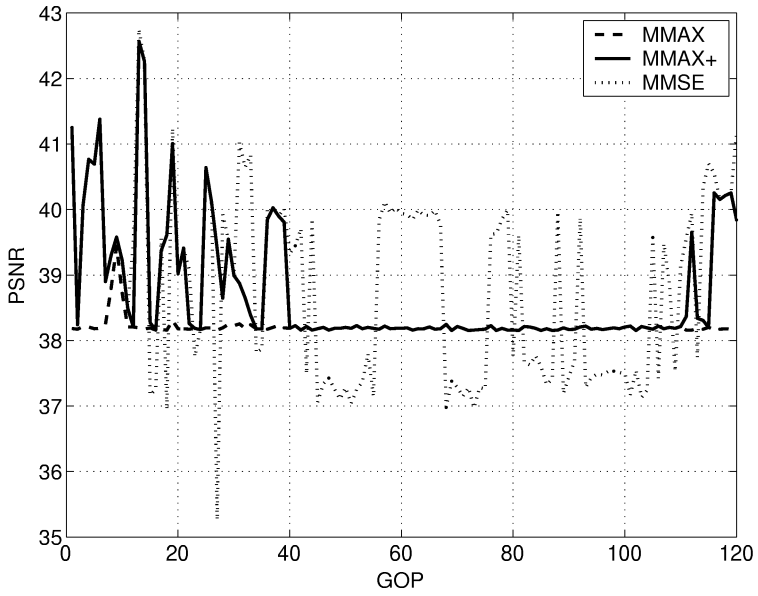
Similar to the CBR transmission, the optimal rate control problem under the MMAX+ criterion can be solved by using a dynamic programming (DP) method [23] or a Lagrangian optimization method [5]. Obviously other techniques are possible to find faster approximate solutions but we provide a result with an optimized DP method to provide a fair comparison between MMAX+ and MMSE solutions. In the DP method which we use in this chapter, given the buffer and channel constraints due to our goal to preserve the MMAX solution, the number of states can be reduced significantly by computing the upper bound of encoding

buffer and TB states. The complexity of the MMSE algorithm is proportional to the number of states (B and TB_{max}) and quantization levels (Q) [48]. Since the EBS and the number of allowable quantization levels are much smaller than B (and TB_{max}) and Q respectively, the complexity to find the optimal solution under a MMAX+ criterion is much lower than that under a MMSE criterion.

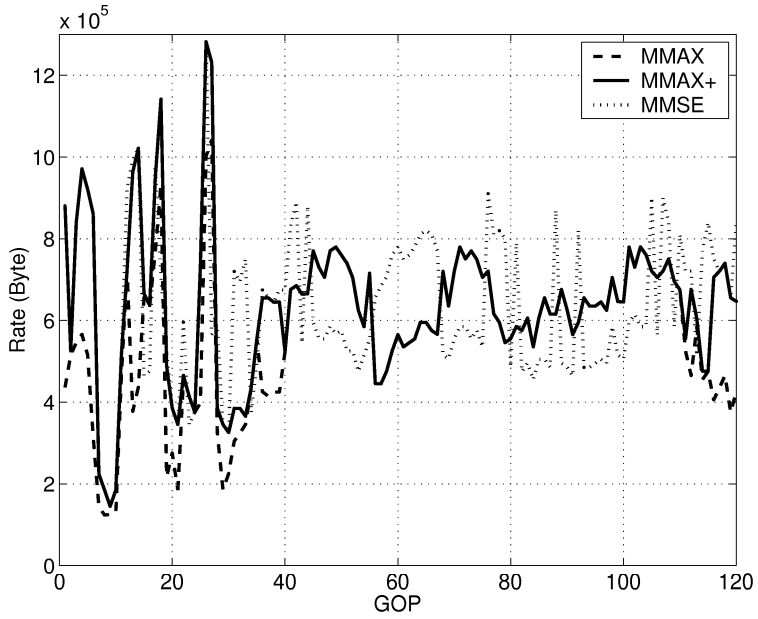
4.4 Experimental results and discussion

In order to verify the performance of the proposed algorithms, we implement these algorithms and test them with 1800 frames from the "Fire Birds" movie sequence. We use the Group of Pictures (GOPs) in MPEG as the basic data unit and the "closed GOP" option in MPEG2 is used to code each GOP independently. To gather the R-D data of the GOPs, each GOP is coded by using 159 different rates roughly between 125 Kbytes and 1705 Kbytes (the difference between steps is roughly 10 Kbytes). The sequence contains 120 GOPs (each GOP has 15 frames) and each GOP is coded by using the MPEG2 TM5 [28] rate control.

Fig. 4.6 shows the optimal solution of MMAX, MMAX+ and MMSE criteria. As shown in the figure (a), the PSNR of MMAX+ is always higher than or equal to the PSNR of MMAX. Also the figure shows that the bit-rate fluctuation among GOPs of the MMSE solution is similar to that of the MMAX and MMAX+ solutions whereas the PSNR fluctuation among GOPs are much larger.



(a)



(b)

Figure 4.6: Comparison of experimental results of CBR transmission. Used channel rate is 10 Mbps (i.e., 625 Kbytes per a GOP interval) and the size of an encoder buffer is 2.5 Mbytes. Therefore the maximum delay used is 4 GOP intervals. Initial and final buffer states are at mid-buffer. (a) and (b) show the PSNR and bit-rate of each GOP respectively.

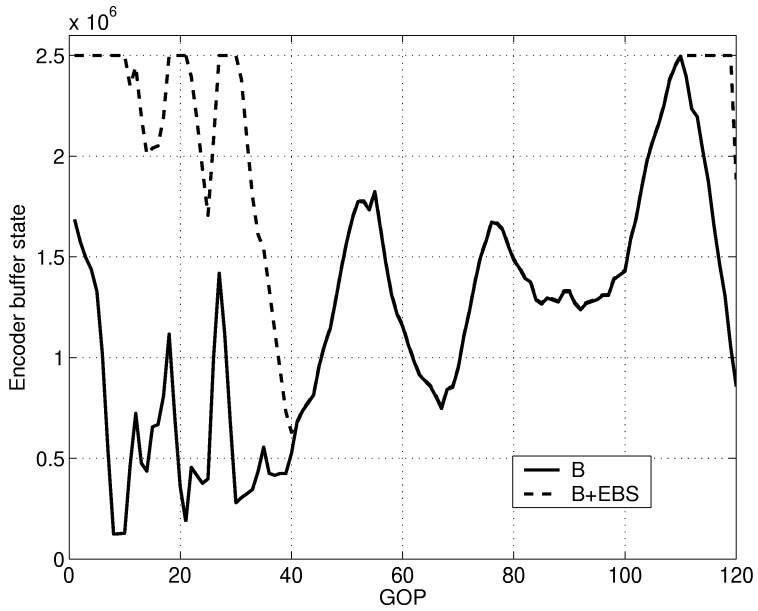


Figure 4.7: Encoder buffer state of CBR transmission. The solid line indicates the encoder buffer state of the MMAX solution and the vertical distance between the dashed and solid lines indicate the effective buffer size (EBS) of each frame.

Fig. 4.7 shows the EBS of each GOP. Note that in the figure, the buffer state of MMAX is always positive since it includes the new coming data (R) at each GOP time. In Fig. 4.7, the EBS of the GOPs between 40 and 110 is zero and this means we cannot increase the bit-rate of these GOPs (otherwise encoder buffer overflow occurs after moving the 110^{th} GOP data into the buffer.) This also explains why the PSNR of the MMAX and MMAX+ solutions of the GOPs between 40 and 110 is identical in Fig. 4.6 (a).

To compare the performance, we also developed an algorithm to find the optimal MLEX solution of the CBR transmission by changing Algorithm 1 slightly

Table 4.1: Performance (PSNR) comparison of proposed MMAX and MMAX+, MMSE and MLEX optimal solutions of CBR transmission. The constraints used are the same as those in Fig. 4.5.

Method	Avg.	Std. Dev.	Min.	Max.
MMAX	38.21	0.137	38.15	39.44
MMAX+	38.60	0.847	38.15	42.57
MMSE	38.72	1.424	35.26	42.75
MLEX	38.56	0.494	38.15	39.44

(i.e., after finding a MMAX solution, instead of terminating the algorithm, keeping the iteration to minimize the 2^{nd} largest distortion and then to minimize the following largest distortion until any distortion cannot be lowered.)

Table 4.1 shows the experimental results of CBR transmission for each criterion. As expected, the minimum PSNR of the MMAX solution is higher than that of the MMSE solution. The MMAX+ criterion improves the average PSNR around 0.4 dB, achieving a value that is near the average PSNR of the MMSE solution. The standard deviation of the MMAX solution shows that the PSNR of each GOP is very similar but the maximum PSNR is relatively high. The reason for this is that some GOPs have simple content and so the PSNR of these GOPs at minimum rate (125 Kbytes) determines the maximum PSNR (see the PSNR and bit-rate of the 9^{th} frame in Fig. 4.6.)

In Table 4.2, the performance of each method is compared for different values of the maximum delay, or equivalently, different sizes of the encoder buffer. The

Table 4.2: Performance (PSNR) comparison of CBR transmission in different maximum delay. The number in the "Method" column indicates maximum delay in GOP interval units. Therefore the sizes of encoder buffers are 5 Mbytes and 1.25 Mbytes respectively. Initial and final buffer states are at mid-buffer (i.e., 2.5 Mbytes and 625 Kbytes respectively.)

Method	Avg.	Std. Dev.	Min.	Max.
MMAX (8)	38.33	0.120	38.27	39.44
MMAX+ (8)	38.59	0.726	38.27	42.57
MMSE (8)	38.73	1.449	35.26	42.75
MLEX (8)	38.56	0.356	38.27	39.44
MMAX (2)	37.71	0.216	37.64	39.44
MMAX+ (2)	38.67	1.242	37.64	42.57
MMSE (2)	38.70	1.441	35.26	42.23
MLEX (2)	38.56	0.716	37.64	40.69

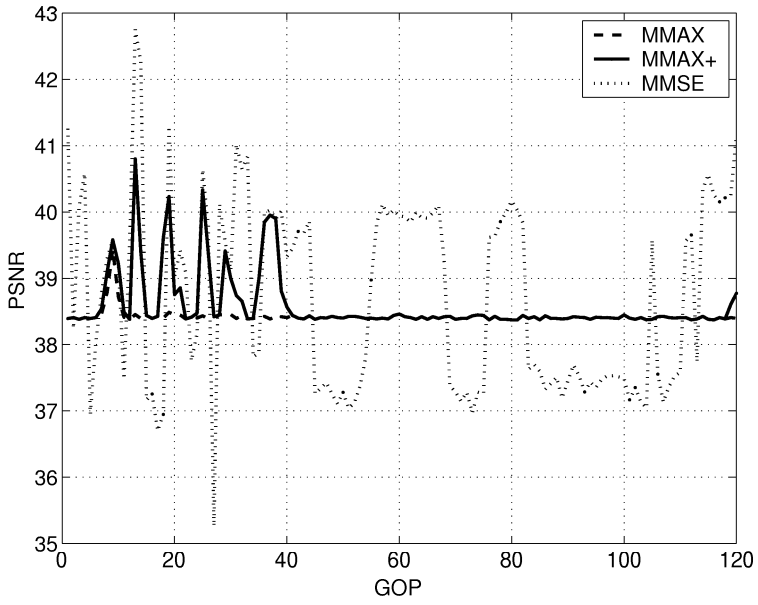
results show that minimum PSNR of the MMAX solution and average PSNR of the MMSE solution are increased (decreased) and the difference of average PSNR between MMAX+ and MMSE solutions is increased (decreased) as the maximum delay is increased (decreased). Because larger buffer size (with increased delay) means that the problem is not as constrained, we can find a better solution (better average PSNR) in a MMSE criterion. Also, if the encoder buffer size is increased then local fluctuation of the bit-rate of GOPs can be absorbed and buffer is not easily overflowed by consecutive complex GOPs. Therefore the minimum PSNR of the MMAX solution is increased and the additional bit-budget available due to the buffer underflow is decreased. Since the remaining bit-budget for the MMAX+ (or MLEX) solution is decreased, the average PSNR cannot

be improved much (and so the average distortion of the MMAX+ and MLEX solutions is almost same when maximum delay is 8.) As a result, the performance of a MMAX criterion is improved (i.e., more bit-budget is used for a MMAX solution) as maximum delay is increased, and therefore the benefits of a MMAX+ criterion are not as significant in this case.

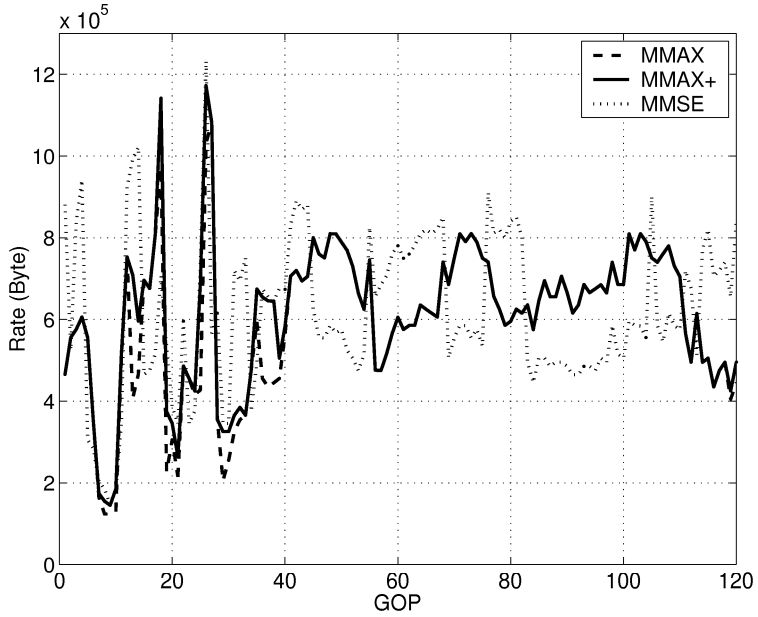
In the other case, if the encoder buffer size is reduced, it is more likely that buffer overflow can occur (i.e., minimum PSNR in MMAX is decreased) and the remaining bit-budget due to buffer underflow is increased (i.e., average PSNR in MMAX+ is increased and the difference of the average PSNR between MMAX+ and MLEX is increased.) Therefore, in this case, the benefits of the MMAX+ criterion are increased and the average PSNR approaches that of the MMSE criterion.

Fig. 4.8 shows the optimal solution of VBR transmission. The result is similar to that of CBR transmission (i.e., the PSNR of MMAX+ is always higher than or equal to the PSNR of MMAX.) Fig. 4.9 (a) and (b) show the EBS and the possible TB state of each frame, respectively. Compared to the EBS of CBR transmission (shown in Fig. 4.7), the EBS of VBR transmission is larger due to the channel flexibility. Note that, in Fig. 4.6 and 4.8, the total amount of the available channel and the maximum delay are same.

Table 4.3 shows the experimental results of VBR transmission for each criterion. The minimum PSNR of the MMAX solution in VBR transmission is higher

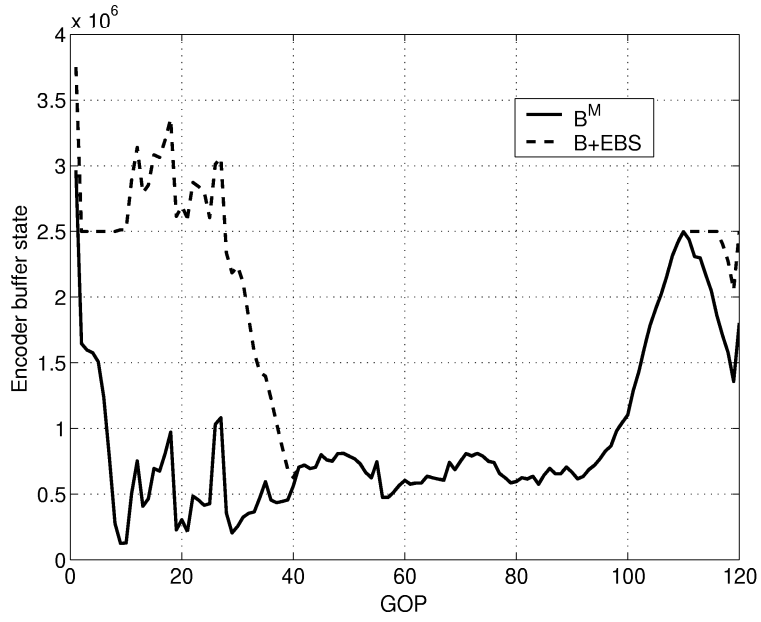


(a)

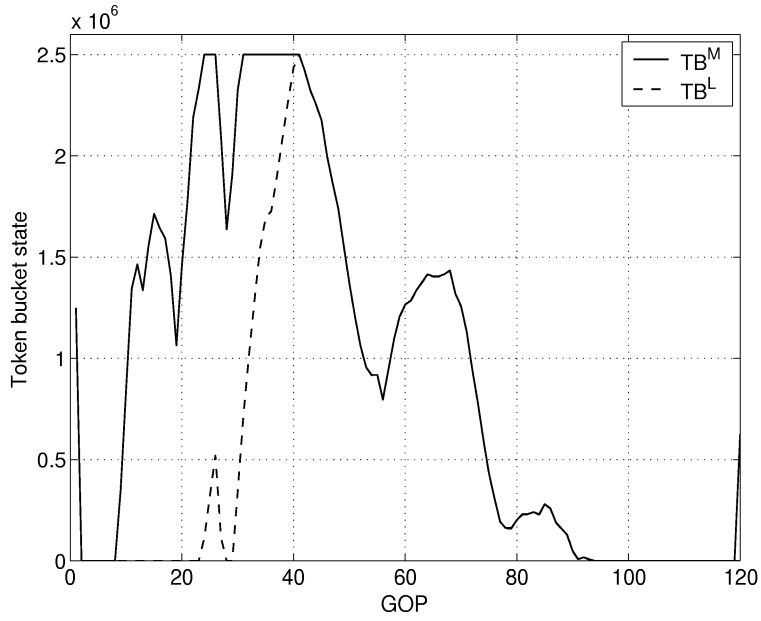


(b)

Figure 4.8: Comparison of experimental results of VBR transmission. Used token rate is 1.25M/sec (i.e., 625K per a GOP interval), the maximum delay is 4 GOP intervals and the size of a TB is 2.5 Mbytes. Initial and final TB and buffer states are at mid-buffer. (a) and (b) show the PSNR and bit-rate of each GOP respectively.



(a)



(b)

Figure 4.9: Encoder buffer state and TB state of VBR transmission. The solid line indicates (a) the buffer state and (b) TB state of the MMAX solution. In (a), the vertical distance between the dashed and solid lines indicate the EBS of each frame. In (b), the dashed line indicates the lower bound of TB state of each frame.

Table 4.3: Performance (PSNR) comparison of the proposed MMAX and MMAX+, and MMSE optimal solutions of VBR transmission. Used constraints are the same as used in Fig. 4.8.

Method	Avg.	Std. Dev.	Min.	Max.
MMAX	38.422	0.140	38.373	39.440
MMAX+	38.583	0.489	38.373	40.800
MMSE	38.733	1.505	35.260	42.753

Table 4.4: Performance (PSNR) comparison of VBR transmission when the maximum delay, TB size and peak rate are changed with respect to the settings of Table 4.3. In the "Method" column, M indicates that the maximum delay is half that in Table 4.3, TB indicates TB size half that in Table 4.3, and P indicates that the peak rate is $1.5 \cdot \bar{C}$. In each case the remaining parameters are not modified.

Method	Avg.	Std. Dev.	Min.	Max.
MMAX (M)	38.312	0.158	38.260	39.440
MMAX+ (M)	38.593	0.804	38.260	42.567
MMSE (M)	38.732	1.502	35.260	42.567
MMAX (TB)	38.280	0.163	38.227	39.440
MMAX+ (TB)	38.590	0.831	38.227	42.567
MMSE (TB)	38.730	1.492	35.260	42.753
MMAX (P)	38.209	0.175	38.153	39.440
MMAX+ (P)	38.599	0.902	38.153	42.567
MMSE (P)	38.724	1.497	35.260	42.753

than that in CBR transmission under the same maximum delay constraint since tokens can be stored in the TB for future use (CBR transmission can be viewed as VBR transmission with TB policing, where TB size is zero.) Although the complexity to find the MMAX+ solution highly depends on the token bucket states of the MMAX solution, in this experiment, the complexity of the MMAX+ algorithm is roughly 20 times lower than that of the MMSE algorithm.

In Table 4.4, the performance of VBR transmission of each method under lower maximum delay, smaller token bucket size, lower peak rate is compared. Because these parameter changes give higher constraints to the problem, the allowable encoder buffer size is reduced as in (4.6) and (4.7) and severe local fluctuation of the bit-rate of GOPs cannot be absorbed. Therefore the minimum PSNR of the MMAX solution is decreased and the additional bit-budget available due to the TB overflow is increased. Since the bit-budget for the MMAX+ solution is increased, the average PSNR can be improved and so the difference of average PSNR between MMAX+ and MMSE solutions is decreased.

4.5 Conclusions

In this chapter, we developed the optimal bit allocation algorithms for CBR and VBR transmission with a token bucket policing function in MMAX and MMAX+ criteria. The MMAX+ criterion is introduced to improve total quality by using

the remaining channel bandwidth under the MMAX criterion. The proposed algorithms lead to an increase in average quality with respect to the MMAX solution, while providing a much more constant and better minimum quality than MMSE solutions. Also algorithms for finding the effective buffer size are proposed. The effective buffer size is used to reduce the number of possible states of each frame and as a result, the complexity of the algorithm to find the optimal MMAX+ solution is reduced.

Chapter 5

Conclusions and Future work

In this thesis, several novel algorithms related to source coding are presented.

At first, a new compression algorithm of digital cameras is proposed, in which the characteristics of a Bayer color CCD array is used to improve coding performance. The simulation shows that the result of the proposed methods outperformed that of the conventional method in a broad range of compression ratios. Because the proposed algorithm uses only around a half size of Y data and requires an additional simple transform, the computing complexity can be decreased. Also, with this algorithm, reducing blocking artifact and fast consecutive capturing can be achieved.

Second, an on-line bit allocation algorithm with a bit-budget constraint is developed. To achieve equal quality of all image frames, a minimizing maximum distortion (MMAX) criterion is used. The future images are estimated from the training data and the images already taken and a "buffer-like" constraint is used

to keep enough memory for the future images. Simulation results show that the performance of this algorithm is close to that obtained by applying an off-line optimal rate control.

Third, an off-line optimal bit allocation algorithm with channel constraints is developed. A MMAX criterion is used to guarantee a minimum quality is achieved and after finding the optimal solution, a MMAX+ criterion is applied to maximally use a given channel and to improve overall quality. CBR and VBR with a token bucket policy are used as a channel and maximum delay is used as a constraint. Also algorithms for finding the effective buffer size are proposed. The effective buffer size is used to reduce the number of possible states of each frame and as a result, the complexity of the algorithm to find the optimal MMAX+ solution is reduced.

5.1 Future work

- **Source coding of captured image:**

  In Chapter 2, we have developed an algorithm to reduce redundancy before compression. By doing this, the complexity of the coding is reduced and also better quality decoded images can be optained under the given bit-budget. The algorithm using "shift" has minimum complexity in terms of the size of source data and the algorithm using "rotation" has near

minimum complexity. Because, in general, image capturing is done with hand-held devices or remote devices, this lower complexity algorithm can provide faster consecutive capturing or low power consuming. The hidden complexity is that this method needs higher decoder power.

Another problem is that trying to achieve better quality if additional complexity is allowed in the encoder side. A high complexity interpolation technique such as directional edge based interpolation [2] [39] [40] gives very good quality reconstructed images but requires several seconds or even several minutes, so it is impossible to use this algorithm in real applications. But if we preserve captured data near lossless then any good interpolation technique can be applied in a decoder side. Therefore the problem is how to simply augment (or interpolate) the captured data to be highly compressed by standard coders while providing near lossless quality.

- **On-line bit allocation with budget constraint:**

  The algorithm of on-line bit allocation with a budget constraint is proposed in Chapter 3. a MMAX criterion is used to achieve near constant quality for all images and the image sequence is considered as i.i.d.

  This algorithm can be extended to a video sequence. For example, a similar algorithm can be used by handling a GOP as an image in the proposed algorithm. But consecutive GOPs in the same scene are highly correlated,

so the estimation of future GOPs can be separated into the estimation of GOPs in the same scene and of GOPs in other scenes. To do this, the length of scene should be estimated for the GOPs in the current scene and average bit-rate of future GOPs should be estimated for the GOPs in the other scene. The result may be improved if bit-budget of scene change is also estimated and applied.

- **Video transmission:**

  In Chapter 4, an off-line optimal bit allocation algorithm in MMAX and MMAX+ criteria is proposed for image or video transmission through CBR and VBR with token bucket channels.

  The next problem to be considered is how to achieve the off-line optimal solution in a MMAX criterion by using an on-line method. Here, we consider the coder that supports scalability property such as SPIHT and JPEG2000 and the encoder buffer that provides data sorting [10]. In this approach, as much coded data as possible is stored and if there is not enough memory to store all images then by using sorting, relatively less important data of the stored images are discarded. Some parts of coded data can then be removed due to scalability property.

Bibliography

[1] J.E. Adams. Design of practical color filter array interpolation algorithms for digital cameras .2. In *Proc. Int. Conf. on Image Proc.*, volume 1, pages 488–492, 1998.

[2] J. Allebach and P.W. Wong. Edge-directed interpolation. In *Proc. Int. Conf. on Image Proc.*, pages 707–710, 1996.

[3] B. E. Bayer. Color imaging array. United States Patent 3,971,065, 1976.

[4] S. K. Biswas and R. Izmailov. Design of a fair bandwidth allocation for VBR traffic in ATM networks. *IEEE/ACM Trans. on Networking*, pages 212–223, April 2000.

[5] J-J. Chen and D. W. Lin. Optimal bit allocation for coding of video signals over ATM networks. *IEEE JSAC*, 15:1002–1015, Aug. 1997.

[6] D. R. Cok. Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. United States Patent 4,642,678, 1987.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, 2001.

[8] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[9] W. B. Davenport. *Probability and Random processes*. McGraw-Hill, 1970.

[10] S. Dolinar, G. Chinn, J. Harel, A. Kiely, M. Klimesh, R. Manduchi, S. Shambayati, M. Vida, A. Ortega, S.-Y. Lee, P. Sagetong, and H. Xie. Region-of-interest data compression with prioritized buffer management. In *Earth Science Technology Conference*, 2001.

[11] W.-C. Feng. *Buffering Techniques for Delivery of Compressed Video in Video-On-Demand Systems*. Kluwer, 1997.

[12] T. R. Fisher, M. W. Marcellin, and M. Wang. Trellis coded vector quantization. *IEEE Trans. on Information Theory*, 37:1551–1566, Nov. 1991.

[13] T. Freeman. Median filter for reconstructing missing color samples. United States Patent 4,724,395, 1988.

[14] Independent JPEG Group. *IJG's JPEG Software Release 6a.* Feb. 1996.

[15] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11:997–1013, Sep. 2002.

[16] C. Herley. A post-processing algorithm for compressed digital camera images. In *Proc. Int. Conf. on Image Proc.*, volume 1, pages 396–400, 1998.

[17] C. Herley. Storage of digital camera images. In *Proc. Int. Conf. on Image Proc.*, Oct. 1999.

[18] D. T. Hoang. *Fast and Efficient Algorithm for Text and Video Compression.* PhD thesis, Brown University, 1997.

[19] D. T. Hoang. Real-time VBR rate control of MPEG video based upon lexicographic bit allocation. In *Data Compression Conference*, pages 374–383, Mar. 1999.

[20] D. T. Hoang, E. L. Linzer, and J. S. Vitter. Lexicographic bit allocation for MPEG video. *J. of Visual Commun. Image Presentation*, 9(4):384–404, Dec. 1997.

[21] D. T. Hoang, E. L. Linzer, and J. S. Vitter. A lexicographic framework for MPEG rate control. In *Data Compression Conference*, pages 101–110, 1997.

[22] P. G. Howard, F. Kossentini, B. Martins, S. Forchhmmer, and W. J. Rucklidge. The emerging JBIG2 standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 8:838–848, 1998.

[23] C.-Y. Hsu, A. Ortega, and A. Reibman. Joint selection of source and channel rate for VBR video transmission under ATM policing constraints. *IEEE Journal on Select Areas in Communications*, 15:1016–1028, Aug. 1997.

[24] K. Illgner, H.-G. Gruber, P. Gelabert, J. Liang, Y. Yoo, W. Rabadi, and R. Talluri. Programmable DSP platform for digital still cameras. In *Int. Conf. Acoustics Speech and Signal Processing*, volume 4, pages 2235–2238, 1999.

[25] ISO/IEC 11172. (MPEG-1): Coding of moving pictures and associated audio - for storage at up to about 1.5 Mbits/s, 1992.

[26] ISO/IEC 13818. (MPEG-2): Generic coding of moving pictures and associated audio information, 1994.

[27] ISO/IEC 14496. (MPEG-4): Information technology - Coding of audio-visual objects, 2002.

[28] ISO/IEC-JTC1/SC29/WG11. Test model 5 (drafts). MPEG93/N0400, 1994.

[29] ITU-T. Video coding for low bitrate communication. ITU-T Recommendation H.263; version 1, Nov. 1995, version 2, Jan. 1998.

[30] K. Jack. *Video Demystified*. LLH Technology Publishing, 1996.

[31] N. Jayant and P. Noll. *Digital Coding of Waveforms*. Englewood Cliffs, NJ:Prentice Hall, 1984.

[32] A. Kaup. Object-based texture coding of moving video in MPEG-4. *IEEE Trans. on Circuits and Systems for Video Tech.*, 9:5–15, Feb. 1999.

[33] R. Kimmel. Demosaicing: Image reconstruction from color CCD samples. *IEEE Trans. on Image Processing*, 4:725–733, June 1995.

[34] C. A. Laroche and M. A. Prescott. Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. United States Patent 5,373,322, 1994.

[35] S.-Y. Lee and A. Ortega. A novel approach of image compression in digital cameras with a bayer color filter array. In *Proc. Int. Conf. on Image Proc.*, volume 3, pages 482–485, 2001.

[36] S.-Y. Lee and A. Ortega. Optimal rate control for video transmission over VBR channels based on a hybrid MMAX/MMSE criterion. In *Proc. Int. Conf. on Multimedia and Expo*, volume 2, pages 93–98, 2002.

[37] S.-Y. Lee and A. Ortega. Optimal rate control for video coding based on a hybrid MMAX/MMSE criterion. In *IS&T/SPIE's 12th International Symposium*, Jan. 2003.

[38] S. Li and W. Li. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *IEEE Trans. on Circuits and Systems for Video Tech.*, 10:725–743, Aug. 2000.

[39] X. Li and M.T. Orchard. New edge directed interpolation. In *Proc. Int. Conf. on Image Proc.*, volume 2, pages 311–314, 2000.

[40] X. Li and M.T. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10:1521–1527, Oct. 2001.

[41] D. W. Lin, M.-H. Wang, and J.-J. Chen. Optimal delayed-coding of video sequences subject to a buffer-size constraint. In *VCIP*, pages 223–234, Cambridge, MA, Nov. 1993.

[42] P. Longère, X. Zhang, P. B. Delahunt, and D. H. Brainard. Perceptual assessment of demosaicing algorithm performance. *Procedings of the IEEE*, 90:123–132, Jan. 2002.

[43] M. W. Marcellin and T. R. Fisher. Trellis coded quantization of memoryless and gauss-markov sources. *IEEE Trans. on Communication*, 38:82–93, Jan. 1990.

[44] M. W. Marcellin, M. A. Lepley, A. Bilgin, T. J. Flohr, T. T. Chinen, and J. H. Kasner. An overview of quantization in JPEG-2000. *Signal Processing: Image Communication*, 17:73–84, Dec. 2001.

[45] A. Ortega. Optimal bit allocation under multiple rate constraints. In *Data Compression Conference*, pages 349–358, Snowbird, Utah, Apr. 1996.

[46] A. Ortega, M. W. Garrett, and M. Vetterli. Rate constraints for video transmission over ATM networks based on joint source/network criteria. In *Annales des Télécommunications*, volume 50, pages 603–616, Jul.-Aug. 1995.

[47] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15:23–50, Nov. 1998.

[48] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal trellis-based buffered compression and fast approximation. *IEEE Trans. on Image Proc.*, 3:26–40, Jan. 1994.

[49] W. Pennebaker and J. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1994.

[50] R. Ramanath, W. E. Snyder, and G. L. Bilbro. Demosaicking methods for bayer color arrays. *Journal of Electronic Imaging*, 11(3):306–315, July 2002.

[51] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG video coder. *IEEE Trans. on Image Proc.*, 3:533–545, Sep. 1994.

[52] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. on Image Proc.*, 2:160–175, Apr. 1993.

[53] A. R. Reibman and B. G. Haskell. Constraints on variable bit-rate video for ATM networks. *IEEE Trans. on Circuits and Systems for Video Tech.*, 2:361–372, Dec. 1992.

[54] J. Ribas-Corbera, P. A. Chou, and S. Regunathan. A generalized hypothetical reference decoder for H.26L. *IEEE Trans. on Circuits and Systems for Video Tech., submitted*, 2001.

[55] A. Said and W. Pearlman. A new, fast and efficient image codec based on set partitioning. *IEEE Trans. on Circuits and Systems for Video Tech.*, 6:243–250, June 1996.

[56] K. Sayood. *Introduction to Data Compression, Second Edition.* Morgan Kaufman Publisher, 2000.

[57] G. Schuster, G.Melnikov, and A. Katsaggelos. A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers. *IEEE Trans. on Multimedia*, 1:3–17, Mar. 1999.

[58] G. Schuster and A. Katsaggelos. The minimum-average and minimum maximum criteria in lossy compression. In *Vistas Astron.*, volume 41, pages 427–437, 1997.

[59] G. Schuster and A. Katsaggelos. Optimal bit allocation depedent quantizers for the minimum maximum distortion criterion. In *Int. Conf. Acoustics Speech and Signal Processing*, volume 4, pages 3097–3100, May. 1997.

[60] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Processing*, 41:3445–3462, Dec. 1993.

[61] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. *Internet Engineering Task Force RFC 2212*, Sep. 1997.

[62] S. Shenker and J. Wroclawski. General characterization parameters for integrated service network elements. *Internet Engineering Task Force RFC 2215*, Sep. 1997.

[63] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizer. *IEEE Trans. on Signal Proc.*, 36:1445–1453, Sep. 1988.

[64] T. Sikora. Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments. *Signal Processing: Image Communication*, 7:381–395, 1995.

[65] A. Skodrs, C. Chistopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18:36–58, Sep. 2001.

[66] P. Sriram and M. W. Marcellin. Image coding using wavelet transforms and entropy-constrained trellis-coded quantization. *IEEE Trans. on Image Processing*, 8:1221–1228, Sep. 1999.

[67] W. Stallings. *High-Speed Networks TCP/IP and ATM Design Principles.* Prentice Hall, 1998.

[68] H. Stark and J. W. Woods. *Probability, Random processes, and Estimation theory for engineers.* Prentice Hall, 1994.

[69] R. Stasidski and J.A. Konrad. New class of fast shape-adaptive orthogonal transforms and their application to region-based image compression. *IEEE Trans. on Circuits and Systems for Video Tech.*, 9:16–34, Feb. 1999.

[70] G. J. Sullivan and R. L. Baker. Rate-distortion optimaized motion compensation for video compression using fixed or variable size blocks. In *Global Telecomm. Conf.*, pages 85–90, 1991.

[71] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Trans. on Image Proc.*, 9:1158–1170, July 2000.

[72] D. Taubman and A. Zakhor. Multirate 3-D subband coding of video. *IEEE Trans. on Image Proc.*, 3:572–588, Sep. 1994.

[73] D. S. Taubman and M. W. Marcellin. *JPEG2000 Image Compression Fundamentals, Standards and Practice.* Kluwer Academic Publisher, 2002.

[74] H. J. Trussell and R. E. Hartwig. Mathematics for demosaicking. *IEEE Trans. on Image Processing*, 11:485–492, Apr. 2002.

[75] K. M. Uz, J. M. Shapiro, and M. Czigler. Optimal bit allocation in the presence of quantizer feedback. In *Int. Conf. Acoustics Speech and Signal Processing*, volume 5, pages 385–388, 1993.

[76] ITU-T VCEG. H.26L buffering ad-hoc group report. http://standard.pictel.com/ftp/video-site/0201-Gen/JVT-B013.doc, Jan. 2002.

[77] ITU-T VCEG. H.26L test model long term number 9 (TML-9) draft0. http://standard.pictel.com/ftp/video-site/h26L, Dec. 2001.

[78] H.-J. Wang and C.-C. Kuo. A multi-threshold wavelet coder MTWC for high fidelity image compression. In *Proc. Int. Conf. on Image Proc.*, pages 652–655, Oct. 1997.

[79] P. H.. Westerink, R. Rajagopalan, and C. A. Gonzales. Two-pass mpeg-2 variable-bit-rate encoding. *IBM Journal of Research and Development*, 43(4), 1999.

[80] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Cambell, and S. K. Mitra. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging h.263 standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 6:182–190, Apr. 1996.

[81] S. G. Wilson. *Digital Modulation and Coding*. Prentice Hall, 1996.

[82] T. Yamada, K. Ikeda, Y. Kim, H. Wakoh, T. Sakamoto, K. Ogawa, E. Okamoto, K. Masukane, K. Oda, and M. Inuiya. A progressive scan ccd image sensor for dsc application. *IEEE Trans. on Solid-State Circuits*, pages 2044–2054, Dec. 2000.

[83] S. Yamanaka. Solid state camera. United States Patent 4,054,906, 1977.