

# Optimal Bit Allocation for Coding of Video Signals over ATM Networks

Jiann-Jone Chen and David W. Lin, *Senior Member, IEEE*

**Abstract**— We consider optimal encoding of video sequences for ATM networks. Two cases are investigated. In one, the video units are coded independently (e.g., motion JPEG), while in the other, the coding quality of a later picture may depend on that of an earlier picture (e.g., H.26x and MPEGx). The aggregate distortion–rate relationship for the latter case exhibits a tree structure, and its solution commands a higher degree of complexity than the former. For independent coding, we develop an algorithm which employs multiple Lagrange multipliers to find the constrained bit allocation. This algorithm is optimal up to a convex-hull approximation of the distortion–rate relations in the case of CBR (constant bit-rate) transmission. It is suboptimal in the case of VBR (variable bit-rate) transmission by the use of a suboptimal transmission rate control mechanism for simplicity. For dependent coding, the Lagrange-multiplier approach becomes rather unwieldy, and a constrained tree search method is used. The solution is optimal for both CBR and VBR transmission if the full constrained tree is searched. Simulation results are presented which confirm the superiority in coding quality of the encoding algorithms. We also compare the coded video quality and other characteristics of VBR and CBR transmission.

**Index Terms**—Asynchronous transfer mode, bit allocation, image coding, optimization methods, quantization, rate distortion theory.

## I. INTRODUCTION

WE consider objective optimization of video coding for ATM networks. Video coding techniques can be classified as being independent or dependent. In information-theoretic terms, independent coding refers to the situation where the distortion–rate (D–R) relation of a later video unit (a picture, a macroblock, etc.) is independent of how an earlier video unit is coded, or in other words, the distortion–rate relations of successive video units are separable, while for dependent coding, it may be dependent, or inseparable. The former includes schemes such as motion JPEG [1]–[3] and certain ways of intraframe coding of macroblocks, while the latter includes those employing motion-compensated predictive coding such as H.26x [4]–[6] and MPEGx [7], [5]. The purpose of our study is twofold: first, to develop efficient algorithms for both independent and dependent video coding,

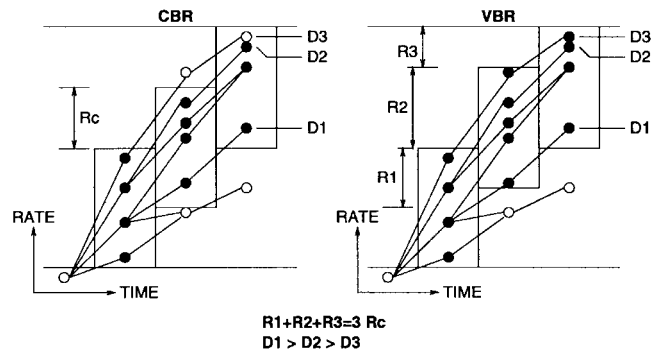


Fig. 1. Constant and variable bit-rate coding.

and second, to compare the properties of optimized CBR (constant bit-rate) and VBR (variable bit-rate) video coding.

Concerning transmission properties, it is widely held that VBR (ATM in particular) is better than CBR due to the time-varying nature in information content of typical video. A simple make-believe example may help illustrate the reason. Consider Fig. 1, which schematically illustrates the different situations one may encounter in CBR and VBR coding of the same four successive video units (numbered as, say, 0, 1, 2, and 3). Each column of dots (open or closed) under the heading CBR or VBR corresponds to a certain video unit. Each series of line segments connecting dots in different columns represents a possible way of coding through the successive video units, with the vertical position of the last dot denoting the total data rate of the coded video.  $D_i$  ( $i = 1, 2, 3$ ) are the distortions associated with three different coding paths. The upper and lower edges of each rectangle delimit the allowed total rate at each time as prescribed by the finite channel transmission capacity and codec buffer sizes. It is seen that, with CBR transmission at a constant rate  $R_c$ , the coding path leading to a final total distortion of  $D_3$  is not allowed because it violates the rate constraints at video unit 2. However, with VBR transmission at an average rate  $R_c$ , it is allowed, leading to a lower total distortion, for at time 2 the transmission rate can be varied momentarily to accommodate it. It is of interest to have some quantitative comparison of VBR and CBR performance under optimized coding, which is part of our goal.

Concerning coding algorithms, Shoham and Gersho [8] proposed an optimal bit allocation algorithm for still-image coding. The algorithm does not address video coding under multiple buffer-imposed constraints, however. For independent coding, Ortega *et al.* [9] considered trellis search (where each

Manuscript received May 15, 1996; revised September 30, 1996. This work was supported by the National Science Council of R.O.C. under Grants NSC 84-2213-E-009-107 and 85-2213-E-009-068. This paper was presented in part at the IEEE International Conferences on Image Processing, Washington, DC, October 1995, and Lausanne, Switzerland, September 1996.

The authors are with Department of Electronics Engineering and Center for Telecommunications Research, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C.

Publisher Item Identifier S 0733-8716(97)04193-0.

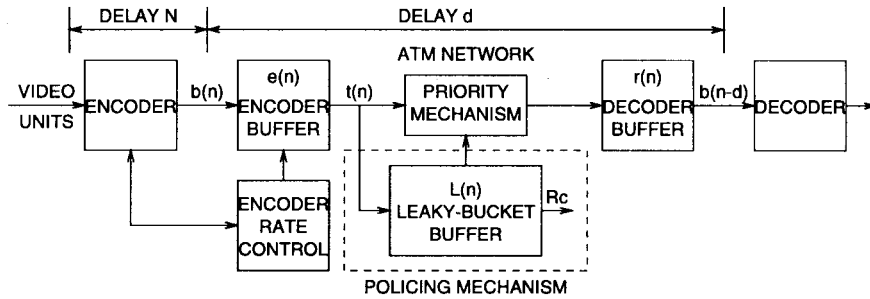


Fig. 2. Codec system with ATM network.

trellis node at stage  $n$  corresponds to the buffer level grown out of a particular sequence of quantizing and coding of the signal up to video unit  $n$ ) for finding the buffer-constrained optimal bit allocation. This method finds the optimal solution at a very high computational load. A closely related work is [10]. Ortega *et al.* [9] also presented several reduced-complexity algorithms which could yield nearly optimal results, among which are some that employ the Lagrange-multiplier technique. However, these Lagrange-multiplier-based algorithms are heuristic. Nor is their optimality characterized theoretically. A different Lagrange-multiplier-based algorithm was considered in [12]. This algorithm makes use of multiple Lagrange multipliers to accommodate the multiple rate constraints. However, its theoretical basis and operational features have not been described clearly. Moreover, the above algorithms were derived largely for CBR transmission. A key result of the present paper is an algorithm (together with the underlying theory) which solves a multiple Lagrange-multiplier problem for independent video coding under the VBR environment of ATM networks. (The VBR environment aside, the multiple-Lagrange-multiplier optimization is a theoretically much more involved problem than the one-Lagrange-multiplier optimization treated in [8] and employed in [9].) We also present some simulation results.

For dependent coding, two alternatives can be envisioned. One is to ignore the dependency in D-R relations, and to employ an independent-coding algorithm to reach a suboptimal solution. The other is to seek an optimal solution taking the D-R dependency into account. We consider the latter in this work. Since the problem is also one of constrained optimization, the Lagrange-multiplier approach can again be attempted. However, it is found to be very cumbersome to use, and the “monotonicity condition” [11] which, if valid, could lead to significant complexity reduction, is found to not always hold in our particular situation. An algorithm based on tree search is thus developed instead.

The coding and transmission system that we consider can be modeled as shown in Fig. 2. As seen, we assume a buffering and transmission delay of  $d$  video units between encoder output and decoder input, where a video unit can be a picture, a macroblock, or some other grouping of picture elements. For simplicity, assume that the encoder and the decoder have infinite processing speed. The encoder performs delayed coding with delay equal to  $N$  video units. With delayed coding, the encoder factors the relative complexity

of present and “future” video units into consideration in its bit allocation and is therefore more likely to yield a better coding quality than considering only one video unit at a time [13, ch. 9].

At time  $n$ , the encoder outputs  $b(n)$  bits (the encoded bits for video unit  $n-N$ ) to the encoder buffer, which outputs  $t(n)$  bits to the ATM network. The ATM network employs a leaky-bucket (LB) policing mechanism. In optimization of video coding, the finite codec buffer sizes and the ATM network policing mechanism place rate constraints on the coded video [14], [16], [17]. More specifically, the decoder buffer fullness and the ATM policing mechanism impose constraints on the allowed transmission rate  $t(n)$  at time  $n$ , which affects the encoder buffer fullness, and in turn imposes constraints on the allowed encoder output rate  $b(n)$  at time  $n$ . Optimization of coding must operate under these constraints. Then a specific transmission rate  $t(n)$  must be determined for each  $n$ . In a sense, the optimization is a joint encoder and channel rate control problem, and we have two control targets, namely,  $b(\cdot)$  and  $t(\cdot)$ .

In what follows, we first derive the rate constraints due to finite channel capacity and finite codec buffer sizes in Section II. These constraints are derived for VBR transmission. But they also apply to CBR transmission in a simplified form. Section III then addresses independent coding. It formulates the optimization problem, and develops an efficient Lagrange-multiplier-based algorithm for solving the problem. Section IV discusses dependent coding, and outlines a tree-search-based solution. Section V presents some simulation results, and Section VI is the conclusion.

## II. OPTIMIZATION CONSTRAINTS

We now derive the rate constraints under which the encoder has to operate. We first consider the constraints on the channel rate  $t(n)$  due to finite decoder buffer and the leaky-bucket ATM policing mechanism, and then that on the encoded bits  $b(n)$  due to the finite encoder buffer and channel rate. Then we explain how these constraints function in coding.

### A. Constraints on the Transmission Bit Rate

As said, constraints on the transmission bit rate are derived from limits of the decoder buffer and the LB buffer.

We first check the rate constraints imposed by the finite decoder buffer size. Let  $R$  denote the decoder buffer size, and let  $r(n)$  be the decoder buffer level after extraction of  $b(n-d)$

[data for the  $(n-d-N)$ th video unit] for decoding. Then

$$r(n) = r(n-1) + t(n) - b(n-d). \quad (1)$$

Cumulated over time, it yields

$$r(n+\ell) = r(n-1) + \sum_{k=n}^{n+\ell} t(k) - \sum_{k=n-d}^{n-d+\ell} b(k) \quad (2)$$

where  $\ell \geq 0$ . To avoid decoder buffer under- and overflows at time  $n+\ell$  [i.e., to ensure  $0 \leq r(n+\ell) \leq R$ ]  $\forall \ell \geq 0$ , therefore,  $t(\cdot)$  must satisfy

$$\begin{aligned} U_D(n, \ell) &\triangleq \sum_{k=n-d}^{n-d+\ell} b(k) - r(n-1) \leq \sum_{k=n}^{n+\ell} t(k) \\ &\leq \sum_{k=n-d}^{n-d+\ell} b(k) - r(n-1) + s \cdot R \\ &\triangleq O_D(n, \ell). \end{aligned} \quad (3)$$

The quantities  $U_D(\cdot, \cdot)$  and  $O_D(\cdot, \cdot)$  define rate boundaries within which decoder under- and overflows, respectively, are avoided. The factor  $s$  will be explained later. Equation (3) summarizes the constraints on transmission rates from the limited decoder buffer size (as well as from the bits of previously encoded video units) up to time  $n+\ell$ . In writing the inequality, we have assumed  $b(n-d)$  and  $r(n-1)$  to be given.

We now move to check the constraints from the network policing function. Let  $L_{\max}$  denote the LB size,  $L(n)$  the LB fullness, and  $R_c$  the average leak rate. Then

$$L(n) = L(n-1) + t(n) - R_c. \quad (4)$$

Cumulatng over time, we obtain

$$L(n+\ell) = L(n-1) + \sum_{k=n}^{n+\ell} t(k) - (\ell+1)R_c \quad (5)$$

where  $\ell \geq 0$ . To ensure that the policing mechanism passes all data intact,  $t(n)$  must be such that the LB never overflows, i.e.,  $L(n) \leq L_{\max} \forall n$  [14]. To fully utilize the channel capacity, we would desire to avoid LB underflows as well, i.e., to let  $L(n) \geq 0 \forall n$ . In this case, the cumulated channel rates are constrained by

$$\begin{aligned} U_L(n, \ell) &\triangleq (\ell+1)R_c - L(n-1) \leq \sum_{k=n}^{n+\ell} t(k) \\ &\leq (\ell+1)R_c - L(n-1) + s \cdot L_{\max} \\ &\triangleq O_L(n, \ell) \end{aligned} \quad (6)$$

where  $\ell \geq 0$  and the role of  $s$  will be explained later.  $U_L(\cdot, \cdot)$  and  $O_L(\cdot, \cdot)$  specify boundary conditions for absence of LB underflow and overflow, respectively.

Equations (3) and (6) together give the constraints on the transmission rate at any time after  $n$ . They depend on the buffer sizes, current buffer fullness, as well as the rates of some previously encoded video units.

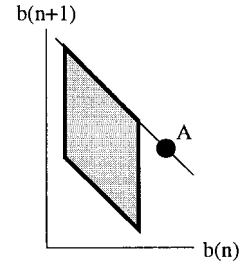


Fig. 3. Typical region of permitted rates when coding delay = 1 video unit.

### B. Constraints on the Encoded Bits

We now derive constraints on encoded bits due to the transmission rate and the finite encoder buffer size. Let  $E$  denote the encoder buffer size, and let  $e(n)$  be the encoder buffer level after the  $(n-N)$ th picture has been coded. With  $b(n)$  and  $t(n)$  as defined earlier, we have

$$e(n) = e(n-1) + b(n) - t(n). \quad (7)$$

Cumulated over time, it yields

$$e(n+\ell) = e(n-1) + \sum_{k=n}^{n+\ell} b(k) - \sum_{k=n}^{n+\ell} t(k) \quad (8)$$

where  $\ell \geq 0$ . To avoid encoder buffer under- and overflow at time  $n+\ell$ , we need  $0 \leq e(n+\ell) \leq E$ , and thus

$$\begin{aligned} \sum_{k=n}^{n+\ell} t(k) - e(n-1) &\leq \sum_{k=n}^{n+\ell} b(k) \\ &\leq \sum_{k=n}^{n+\ell} t(k) - e(n-1) + s \cdot E \end{aligned} \quad (9)$$

which constrains the cumulated number of encoded bits at all future time under a given sequence of transmission rates  $\{t(k), k = n, n+1, \dots\}$ .

The scale factor  $s$  is introduced in (3), (6), and (9) for the purpose of rate control, so that a target buffer fullness less than  $L_{\max}$  and  $E$  can be set for  $\ell = N$  (the conclusion of the  $N+1$  delay-coded video units); that is,  $s = 1$  for  $\ell < N$  and  $s < 1$  for  $\ell = N$ . For example, we may choose  $s$  to be less than, equal to, or greater than  $\frac{1}{2}$  at  $\ell = N$  according to whether the video units after  $n$  are expected to be more complex, equal in complexity, or less complex than the video units  $[n-N, n]$  that are being delay encoded.

The constraints (9), with given  $\{t(k)\}$ , mark out a region of permissible values for the  $N+1$  vector  $[b(n), \dots, b(n+N)]$  in the  $N+1$ -dimensional space. For example, for  $N = 1$ , a typical region may look like the shaded area shown in Fig. 3. For signals with convex D-R relations, the optimal solution should situate on or near the top border of the region.

### C. Interpretation of the Rate Constraints

Fig. 4 summarizes the relation among constraints (3), (6), and (9) at time  $n$ , when we begin the encoding of video units  $[n-N, n]$ . For convenience, we draw the case for  $d > N+1$ . At time  $n$ , the encoded bits and the LB fullness are available for time up to  $n-1$ . (They are already fixed and cannot be

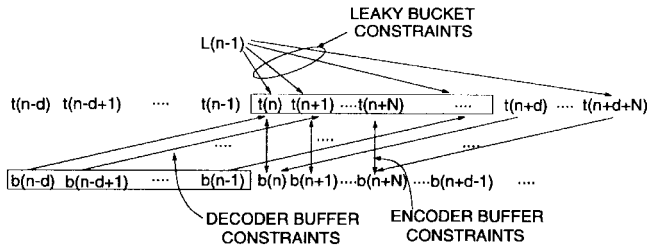


Fig. 4. Relation among rate constraints.

changed.) The bits  $\{\sum_{k=n-d}^{n-d+\ell} b(k), \ell = 0, \dots, d-1\}$  and  $L(n-1)$  place constraints on  $\sum_{k=n}^{n+\ell} t(k), \ell = 0, \dots, d-1$  and the constraints are unidirectional. The relation between  $\{b(k), k \geq n\}$  and  $\{t(k), k \geq n\}$  is different. The situation can be appreciated by looking at Fig. 2. While the allowed transmission rates  $\{t(k), k = n, \dots, n+N\}$  [as determined from (3) and (6)] constrain  $\{b(k), k = n, \dots, n+N\}$  as in (9), after the encoding of video units  $[n-N, n]$  (which determines  $\{b(k), k = n, \dots, n+N\}$ ), specific values for  $\{t(k), k = n, \dots, n+N\}$  must be determined to facilitate transmission. For this,  $\{b(k), k = n, \dots, n+N\}$  now act as constraints through the finite encoder buffer size. From this perspective, (9) can be rewritten into

$$\begin{aligned} O_E(n, \ell) &\triangleq \sum_{k=n}^{n+\ell} b(k) + e(n-1) - E \leq \sum_{k=n}^{n+\ell} t(k) \\ &\leq \sum_{k=n}^{n+\ell} b(k) + e(n-1) \triangleq U_E(n, \ell) \end{aligned} \quad (10)$$

where  $\ell \geq 0$  and  $O_E(\cdot, \cdot)$  and  $U_E(\cdot, \cdot)$  specify boundaries beyond which encoder buffer over- and underflow, respectively, would occur. Therefore, the constraining relationship between  $\{t(k), k = n, \dots, n+N\}$  and  $\{b(k), k = n, \dots, n+N\}$  is bidirectional, with both directions at work at different times during the coding and transmission process. Moreover,  $\{b(k), k = n, \dots, n+N\}$  and  $\{t(k), k = n+d, \dots, n+d+N\}$  are related through the decoder buffer relationship (2). Hence, the former are constrained by the latter in the following way:

$$\begin{aligned} \sum_{k=n}^{n+\ell} t(k) + r(n-1) - R \\ \leq \sum_{k=n-d}^{n-d+\ell} b(k) \leq \sum_{k=n}^{n+\ell} t(k) + r(n-1) \end{aligned} \quad (11)$$

for  $\ell = d, \dots, d+N$  (a reformulation of (3) for these values of  $\ell$ ). The constraints are again unidirectional for the coding of video units  $[n-N, n]$ . (They will work in the other direction in the coding of later video units.)

In summary, in coding video units  $[n-N, n]$ , the effective constraints are (3) for  $\ell \in [0, d]$ , (6) for  $\ell \in [0, d+N]$ , (9) and (10) for  $\ell \in [0, N]$ , and (11) for  $\ell \in [d, d+N]$ . Under constraint are  $d+2N+2$  quantities, namely,  $\{t(k), k = n, \dots, n+d+N\}$  and  $\{b(k), k = n, \dots, n+N\}$ .

### III. OPTIMAL INDEPENDENT CODING—THE LAGRANGE MULTIPLIER APPROACH

We now consider optimal coding in the case where the video units are independently coded, examples being motion JPEG and certain ways of intraframe coding. Consider an additive distortion measure, that is, the encoder seeks to conduct the following minimization:

$$\min_{Q(m), n-N \leq m \leq n} \sum_{m=n-N}^n D(m) \quad (12)$$

where  $Q(m)$  denotes all possible ways of coding video unit  $m$  and  $D(m)$  denotes the corresponding distortion in this video unit, subject to the constraints on  $\{b(m), m = n, \dots, n+N\}$  in (9) together with (3) and (6). The choice for  $D(m)$  is limitless, of which the MSE and the weighted MSE are two common examples.

As discussed, we approach this problem by way of Lagrange-multiplier optimization. The Lagrange-multiplier method only finds solutions on the convex hull of a D-R relation, while it is known that actual D-R relations for real video may be nonconvex. Hence, it is possible that the Lagrange-multiplier solution is suboptimal, and the truly optimal solution lies somewhere between it and the rate-region boundary. A search procedure can be appended to the Lagrange-multiplier method to look for the truly optimal solution [8], but it is deemed too complicated to implement and may lead to small improvement only.

#### A. Suboptimal Control of the Transmission Rate

For the present work, it appears overwhelming to consider all possible bit allocations afforded by the rate constraints derived in the last section (and illustrated in Fig. 4) in the Lagrange-multiplier optimization. We take a handwaving measure, and employ a kind of greedy method to control the transmission rate  $t(\cdot)$ , but leave  $b(\cdot)$  under the full freedom of the constraints that remain. More specifically, the rates are controlled in the following way.

Intuitively, we should maximize  $\sum_{k=n}^{n+N} b(k)$  in hope of the best video quality. From (9), this calls for choosing the maximum possible value for  $\sum_{k=n}^{n+N} t(k)$ , for then, the allowed range for  $\sum_{k=n}^{n+N} b(k)$  is the highest among all possible. From (3) and (6),  $\sum_{k=n}^{n+N} t(k)$  should not exceed  $\min\{O_L(n, N), O_D(n, N)\}$ . Thus, the rate budget  $R_b$  for  $\sum_{k=n}^{n+N} b(k)$  is set to be

$$\begin{aligned} \min\{(O_L, O_D)(n, N)\} - e(n-1) &\leq R_b \\ &\leq \min\{(O_L, O_D)(n, N)\} - e(n-1) + s \cdot E \end{aligned} \quad (13)$$

[where for notational convenience we write  $(f_1, \dots, f_n)(\cdot)$  as a shorthand for  $f_1(\cdot), \dots, f_n(\cdot)$ ] and the allocated total rate should be as close to the right-hand side (RHS) as possible. In determining the bit allocation for the video subsequence  $[n-N, k]$  (where  $k < n$ ), we allow  $t(\cdot)$  to be as large as possible to avoid encoder buffer overflow or to be as small as possible to avoid encoder buffer underflow up to  $k$ , without regard for how this would affect later coding. (Step S3 in the algorithm presented later in this section is an embodiment of

this principle.) After encoding of video units  $[n - N, n]$ , we determine  $\{t(k), k = n, \dots, n + N\}$  according to

$$\begin{aligned} & \max\{0, (U_D, U_L, O_E)(n, \ell)\} \\ & \leq \sum_{k=n}^{n+\ell} t(k) \leq \min\{(O_D, O_L, U_E)(n, \ell)\} \quad (14) \end{aligned}$$

where  $\ell = 0, 1, \dots, N$ . Results from several specific choices of  $\sum_{k=n}^{n+\ell} t(k)$  are presented later.

The above method of rate control is acknowledgeably suboptimal. We leave fully optimal solutions to potential future work. (We remark that this is an issue only for VBR transmission. For CBR transmission, the channel rate is fixed, and there is no issue of suboptimality.)

### B. Lagrange-Multiplier Optimization

With greedy determination of  $t(\cdot)$ , only  $b(\cdot)$  are subject to optimization. Applying the Lagrange-multiplier technique to (12), we obtain

$$\begin{aligned} \min_{Q(m), n-N \leq m \leq n} & \left\{ \sum_{m=n-N}^n D(m) + \lambda_0 b(n) \right. \\ & \left. + \lambda_1 \sum_{m=n}^{n+1} b(m) + \dots + \lambda_N \sum_{m=n}^{n+N} b(m) \right\} \quad (15) \end{aligned}$$

where  $\lambda_i, i = 0, 1, \dots, N$  are the Lagrange-multipliers. Term  $[\lambda_0, \lambda_1, \dots, \lambda_N]$  a Lagrange-multiplier vector. To find the optimal solution, we will have to step through a series of candidate Lagrange-multiplier vectors. For each given vector, the above minimum can be evaluated if we have the D-R relation for video units  $n - N$  to  $n$ . Therefore, the first task in Lagrange-multiplier optimization is "data generation" (using the term of Ramchandran *et al.* [11]) which computes the D-R relation for each macroblock in video units  $n - N$  to  $n$ . Next, we have to find the optimal Lagrange-multiplier vector.

Optimization under multiple constraints (hence multiple Lagrange multipliers) can be very difficult unless the problem possesses some simplifying structure. Fortunately, we have such a situation: the multimultiplier problem (15) can be broken down into a series of one-multiplier problems, each of which is readily solvable using the technique of [8]. The derivation of this result takes some theoretical excursion, and is relegated to the Appendix. Below, we describe the ensuing algorithm. It has a recursive construct.

First, we freeze all Lagrange multipliers except  $\lambda_N$  to zero and solve (15) subject to the constraint (13) on the total rate. This is a one-Lagrange-multiplier problem which is readily solvable using the method of [8]. The solution will satisfy the total-rate constraint for video units  $[n - N, n]$ , but not necessarily at any time between  $n - N$  and  $n$ . In the case  $N = 1$ , for example, the solution may lie along the line that defines the top border of the allowed rate region in Fig. 3, such as point A, assuming the D-R relations are convex. We thus have to check for possible violation of the "nonterminal" rate constraints. If the constraints are satisfied everywhere, then we have obtained the optimal bit allocation. Otherwise, some

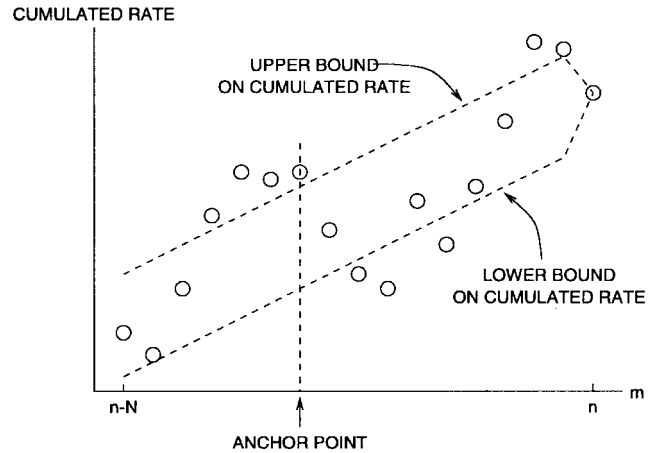


Fig. 5. Concept of anchor points. Abscissa denotes time (index of video units). Ordinate positions of circles give cumulated bit allocation for video units  $[n - N, m]$ . For simplicity, the upper and lower bounds on cumulated rate are drawn as straight lines. This is the case for CBR transmission. For VBR transmission, they may be jagged.

other solution has to be found. For the  $N = 1$  example above, the true solution should lie in the upper right corner of the allowed rate region.

To obtain the true solution, we first identify the *anchor point* in video sequence  $[n - N, n]$ , defined as the last video unit that violates the RHS (overflow) constraint in (9) that comes before the first violation of the left-hand side (LHS) (underflow) constraint, or the last video unit that violates the LHS constraint in (9) that comes before the first violation of the RHS constraint, whichever condition holds. If only one kind of violation occurs over the sequence, then it is the last video unit before  $n$  where violation occurs. The idea of anchor points is illustrated in Fig. 5. Denote the anchor point by  $v$ . Define the video subsequence  $[n - N, v]$  as the *anchor subsequence*. Now, we can modify the bit allocation to address the violations.

The basic idea is as follows. Conduct an optimal bit allocation for the anchor subsequence subject only to the total-rate constraint on this subsequence (a one-Lagrange-multiplier optimization problem). Examine whether all of the nonterminal rate constraints for  $[n - N, v]$  are satisfied. If not, then identify the anchor point in  $[n - N, v]$  in the same manner as we have done for the overall sequence  $[n - N, n]$ , set  $v$  to denote this new anchor point, and recursively descend into the new anchor subsequence as we have done in the case of the initial anchor subsequence above. If yes, then we have obtained the optimal bit allocation for  $[n - N, v]$ . For "sliding-window coding" in which video units  $[n - N + 1, n]$  are merely employed for bit allocation but only video unit  $n - N$  is actually encoded and sent to buffer at each time  $n$ , we may stop here. For "jumping-window coding" with jump distance  $N + 1$ , in which all video units  $[n - N, n]$  are actually encoded at time  $n$  and buffered for later transmission, we continue with the following: subtract from  $R_b$  the rate allocated to the last anchor subsequence, advance to the subsequence  $(v, n]$ , and repeat the above procedure to allocate the remaining bits to it.

To effect the one-constraint optimal bit allocation over an anchor subsequence, we freeze all Lagrange multipliers except

$\lambda_{v-n+N}$  to zero, and conduct the minimization of (15) subject to the constraint (9) for  $\ell=v-n+N$  only, with the additional requirement that the solution should lie on or near the LHS or the RHS rate boundary at  $\ell=v-n+N$ , depending on which boundary is exceeded at the anchor point. We remark that, due to the recursive nature of the overall procedure, the whole delay-encoded video sequence  $[n-N, n]$  can be viewed as the startup anchor subsequence.

As an example, consider again the  $N=1$  example shown in Fig. 3 where the initial optimization subject only to (13) yields a solution at point A. The anchor point is thus  $n-1$ , and the anchor subsequence is  $[n-1, n-1]$ . Optimized coding of the anchor subsequence subject to the RHS of (9) for  $\ell=0$  would yield a solution along the right border of the rate region. The desired ‘‘corner solution’’ is obtained by allocating the remaining bits to  $n$  and optimizing its coding subject to this rate constraint.

We summarize the foregoing Lagrange-multiplier optimization procedure into an algorithmic description.

*Algorithm 1:* S1: Compute constraints on the total bit budget  $R_b$  for the  $N+1$  video units using (13). Set  $\text{violation}=\text{overflow}$ ,  $n_a = n-N$ ,  $n_b = n$ , and  $R_1 = R_b$ .

S2 (Trial optimization): Employ a Lagrange multiplier  $\lambda$  to perform the optimization

$$\min \left\{ \sum_{m=n_a}^{n_b} D(m) \right\} \text{ subject to} \quad (16)$$

$$\begin{cases} \sum_{m=n_a}^{n_b} b(m) \leq R_1 & \text{if } \text{violation} = \text{overflow}, \\ \sum_{m=n_a}^{n_b} b(m) \geq R_1 & \text{if } \text{violation} = \text{underflow} \end{cases}$$

(where the inequalities should be satisfied as close to being equality as possible). Obtain the associated bit allocation  $\{b(k), k = n_a, \dots, n_b\}$ .

S3: Check (9), (3), and (6) for any buffer under- or overflows under the above bit allocation. In their absence, go to S4. Otherwise, locate the anchor point  $v$  in  $[n_a, n_b]$  as defined previously. Set the variable  $\text{violation}$  to indicate the kind of violation. Set bit budget

$$R_1 = \begin{cases} \min\{(O_L, O_D)(n_a, v)\} + E - c(n_a - 1) \\ \max\{(U_L, U_D)(n_a, v)\} - c(n_a - 1) \end{cases}$$

$$\text{if } \text{violation} = \begin{cases} \text{overflow} \\ \text{underflow}. \end{cases}$$

for video units  $[n_a, v]$ . Set  $n_b = v$  and go to S2.

S4: Determine  $\{t(m), m = n_a, \dots, n_b\}$  according to (14), and update the buffer levels  $r(n_b)$ ,  $c(n_b)$ , and  $L(n_b)$  using (1), (4), and (7).

If  $(n_b = n_a)$  go to S5; else let  $R_1 = R_b - \sum_{m=n-N}^{n_b} b(m)$ ,  $n_a = n_b + 1$ , and  $n_b = n$  and go to S2.

S5: Output  $b(k)$  and  $t(k)$  for  $k = n-N, \dots, n$ . Increment  $n$  by  $N+1$ , and go to S1.  $\square$

The above algorithm applies to jumping-window coding with jump distance  $N+1$ . Modification of it for sliding-window coding is easy and omitted.

### C. Complexity Analysis

The above algorithm contains two phases: a data generation phase and a Lagrange-multiplier optimization phase. Assume that in the data generation phase, we obtain the D–R relation of each video unit in  $[n-N, n]$ . In addition, the ‘‘singular Lagrange multiplier values’’ [8] for these video units are also obtained and sorted. (A singular Lagrange multiplier for video unit  $m$  is a number  $\lambda$  for which there is more than one solution to the problem  $\min\{D(m) + \lambda b(m)\}$ , i.e., more than one way of quantizing the video unit to yield the minimum. It is equal to the slope of a line segment on the convex hull of the D–R relation of the video unit.)

For the Lagrange-multiplier optimization, assume there are  $\Lambda$  singular Lagrange multipliers per video unit.  $\Lambda$  is upper bounded by the product of the number of macroblocks in each video unit and the number of selectable quantizers for each macroblock. In the above-presented algorithm, the most computation-intensive step is S2 (trial optimization). In comparison, other steps have negligible complexity. Consider first a worst case scenario where the trial optimization always results in constraint violations (except when  $n_b = n_a$ ), and the new anchor point always occurs at  $n_b - 1$ . It then takes  $N+1$  trial optimizations on anchor subsequences of lengths  $N+1, N, N-1, \dots, 1$ , respectively, to obtain optimal bit allocation for the video unit  $n-N$ . For sliding-window coding, this is when we slide the window and begin optimization for the next segment of video. For jumping-window coding, we continue optimization for video units  $[n-N+1, n]$ , which requires  $N$  trial optimizations to attain the optimal solution for video unit  $n-N+1$ , and so on.

A pass over S2 requires, at worst, search among  $(n_b - n_a + 1)\Lambda$  singular Lagrange multipliers for solution, at a complexity on the order of  $\log_2[(n_b - n_a + 1)\Lambda]$  steps employing logarithmic search. Each search step requires up to the order of  $n_b - n_a + 1$  arithmetic operations to compute the total rate and compare it against the constraint. For sliding-window coding, therefore, the worst case complexity [denoting it by  $W_s(N+1)$ ] is on the order of

$$\begin{aligned} W_s(N+1) &= \sum_{k=1}^{N+1} k \log_2(k\Lambda) \\ &= \sum_{k=1}^{N+1} k \log_2 \Lambda + \sum_{k=1}^{N+1} k \log_2 k \\ &< \sum_{k=1}^{N+1} k \log_2 \Lambda + \int_1^{N+2} x \log_2 x \, dx \\ &= \frac{(N+1)(N+2)}{2} \log_2 \Lambda \\ &\quad + \frac{(N+2)^2}{2} \log_2(N+2) \\ &\quad - \frac{(N+1)(N+3)}{4 \cdot \ln 2} \end{aligned} \quad (17)$$

per video unit. This worst case scenario may appear too pessimistic. However, we have too few data to define a statistically meaningful majority or average situation. Nevertheless, it seems heuristically reasonable to consider the following

more favorable situation than the worst case, namely, that the anchor subsequence reduces geometrically in length with each recursion. We can derive (derivation omitted) that the complexity in this case, denoted  $G_s(N+1)$ , is an order of magnitude lower in power of  $N$  than  $W_s(N+1)$  as

$$G_s(N+1) = \mathcal{O}\left(\frac{N+1}{1-f}[\log_2 \Lambda + \log_2(N+1)]\right) \quad (18)$$

per video unit, where  $f$  is the factor by which the lengths of anchor subsequences reduce in each recursion.

For jumping-window coding, the worst case complexity is given by

$$W_j(N+1) = \sum_{m=1}^{N+1} W_s(m) = \mathcal{O}\left(\frac{N^3}{6}[\log_2 \Lambda + \log_2 N]\right) \quad (19)$$

for the whole sequence  $[n-N, n]$ . Dividing it by  $N+1$ , we obtain the average complexity per video unit, which is the same in order of magnitude as  $W_s(N+1)$  except for a three-times lower proportionality constant. For the more favorable situation of geometrical shrinkage of anchor subsequence, a bound on complexity is given by

$$\begin{aligned} \bar{G}_j(N+1) &= \sum_{m=1}^{N+1} G_s(m) \\ &= \mathcal{O}\left(\frac{N^2}{2(1-f)}[\log_2 \Lambda + \log_2 N]\right) \end{aligned} \quad (20)$$

for the whole sequence  $[n-N, n]$  or  $1/N$  of it per video unit. If, in addition, we assume that the final set of anchor subsequences for which the optimal solutions of S2 satisfy all rate constraints is also geometrically distributed in length, then the complexity for jumping-window coding is brought down further by one order of magnitude in power of  $N$  to

$$G_j(N+1) = \mathcal{O}(cN[\log_2 \Lambda + \log_2 N]) \quad (21)$$

where  $c$  is a constant, for the whole sequence  $[n-N, n]$  and  $1/N$  thereof per video unit—which is lower by an order of magnitude in power of  $N$  than sliding-window coding.

Work is in progress concerning an algorithm which guarantees geometrical convergence regardless of how anchor subsequences shrink in recursion [15].

#### IV. OPTIMAL DEPENDENT CODING—CONSTRAINED TREE SEARCH

We now direct attention to optimal bit allocation for coding schemes where the D–R relation of a later video unit may depend on how an earlier unit is encoded. Examples of such schemes include ITU-T's H.26x series recommendations and ISO's MPEGx standards.

In common video coding schemes, the distortion and rate of a video unit are both controlled by choice of the quantizer step size. In the case of dependent coding, different choices of the quantizer for an earlier video unit may lead to different D–R relations for later video units. Hence, the collection of all possible coding choices for a video sequence, say

$[n-N, n]$ , can be organized into a tree. The first-level nodes are defined by the quantizer choices for unit  $n-N$ . Each node has as many children (second-level nodes) as there are choices for the quantizer. Each tree branch is associated with a certain distortion and a certain data rate. In the case of independent coding, the tree degenerates into a trellis because successive levels' distortion–rate relations are independent. Such a trellis is employed in solving for the constrained optimal bit allocation in independent video coding in [9] and [10]. Use of the tree in dependent coding is considered in [11], where the tree is also termed a trellis.

Several methods exist for finding the optimal solution. One is still using a Lagrange multiplier approach. But since the D–R relations of successive video units are not separable, performance of the minimization (15) for a given set of Lagrange multiplier values can no longer be accomplished by minimizing for each video unit individually and then combining the results, but entails a search over all of the terminal nodes of the above D–R tree for the lowest value of (distortion +  $\lambda$  · rate). Aside from the issue of tree search under given multiplier values, another issue is the number of multiplier values that need be searched to find a solution at the desired constraint boundary. This number can be minimized using a logarithmic-type search scheme, such as the hierarchical method in [16].

To reduce the tree-search complexity for given Lagrange multiplier values, a kind of “monotonicity condition” in the D–R relations of successive video units was assumed and exploited in [11]. Briefly, this condition says that a video unit will have a worse D–R relation if its preceding video unit is quantized with a coarser step size than if with a finer step size, in the sense that the D–R relation of the latter lies toward the upper right of that of the former. While this is experimentally confirmed in some cases [11], we have found that the condition may not always hold for *any* conceived way of coding under the H.26x or the MPEGx framework. Some examples are shown in Fig. 6. The figure plots the operational D–R relation of the second picture in two successive predictive-coded pictures under H.261 framework.  $q1$  and  $q2$  denote the average quantizer step sizes used in the encoding of the previous and the present pictures, respectively. Under H.261, there are many ways to encode a picture to yield the same average quantizer stepsize. Points in the figure correspond to that minimizing the distortion in the respective pictures. Note that a segment of the curve for  $q1 = 4$  clearly lies to the lower left of the curve for  $q1 = 2$ , violating monotonicity.

The above violation of monotonicity can probably be attributed in part to the fact that, in “optimizing” the quantization vector, we only minimized the distortion, but paid no attention to its rate implications. But this is at least one reasonable (although suboptimal) way to trim the full coding tree for delayed coding over a window of several pictures—the tree size could be astronomical without trimming. (Alternatively, one could conceive of the existence of a problem whose solution set observes the structure of this trimmed tree. In this case, the problem is endowed with inherent nonmonotonicity.) Another possible cause of nonmonotonicity, under H.261 and similar coding frameworks, is the allowed use of intraframe

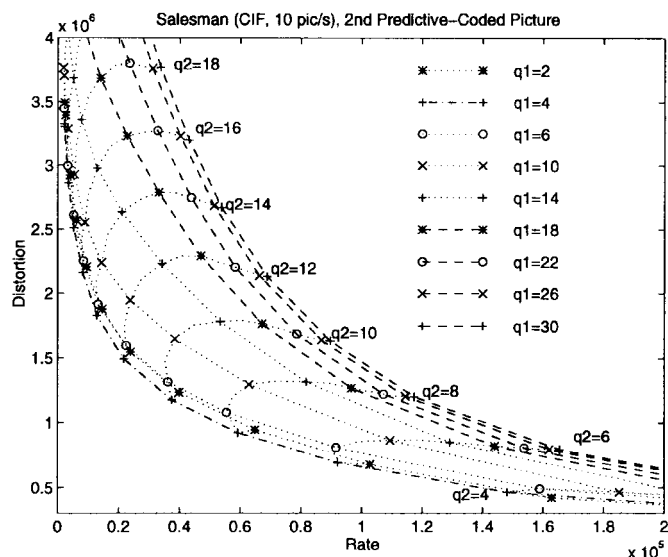


Fig. 6. Example showing that monotonicity may not always hold.

coding for macroblocks in a predictively coded picture. If the reference picture is coarsely quantized, the encoder may choose to use intraframe coding for the present picture, which may result in a lower distortion at the expense of more bits for the present picture when compared to the case where the reference picture is more finely quantized. This already could result in violation of the monotonicity condition. Adding to it is that the variation in rate and distortion from the above phenomenon could be irregular to sum up to more conspicuous violations.

Nevertheless, one should note that the above in no way nullifies the usefulness of the monotonicity assumption in reduction of complexity for tree/trellis search. The reduced-complexity algorithm so obtained can be applied even in the face of nonmonotonicity to arrive at a suboptimal solution. However, in the present work, we opt to avoid this additional source of potential suboptimality. In this case, with the dependency in D-R relations and without the monotonicity condition, the Lagrange multiplier method appears too tedious and inefficient. (The Appendix touches on this briefly.) An attempt was made to simplify the problem by disregarding some constraints in (3), (6), and (9), in particular, those restricting encoder buffer underflows. Simulation results show that this, in many cases, increases the distortion in the coded video because these underflows do occur, and the ensued bit stuffing nibbles away channel bandwidth that could otherwise be used for video data, although the increase in distortion may not always be significant.

In conclusion, we deemed the Lagrange multiplier approach too unwieldy in dependent coding, and we opted for straightforward search over the coding tree under the previously derived rate constraints. The search can be carried out recursively by growing the tree one level at a time (corresponding to quantization and coding of one successive video unit using all possible quantizer step sizes), trimming the new-grown branches according to the previously derived rate constraints, and recording the distortion associated with each survivor path, until we come to the end of the  $N+1$  video

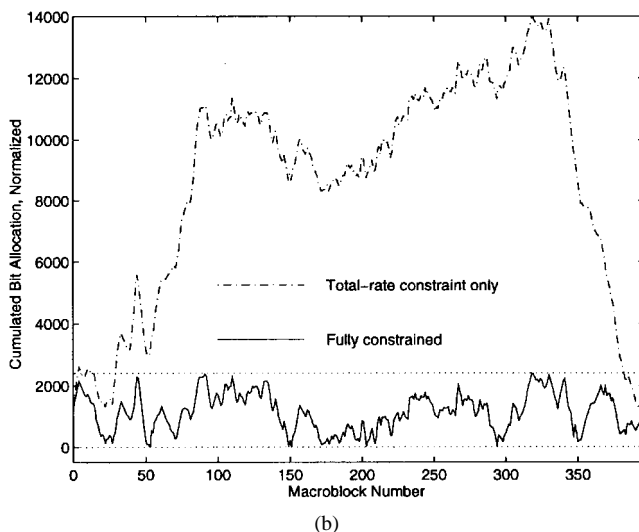
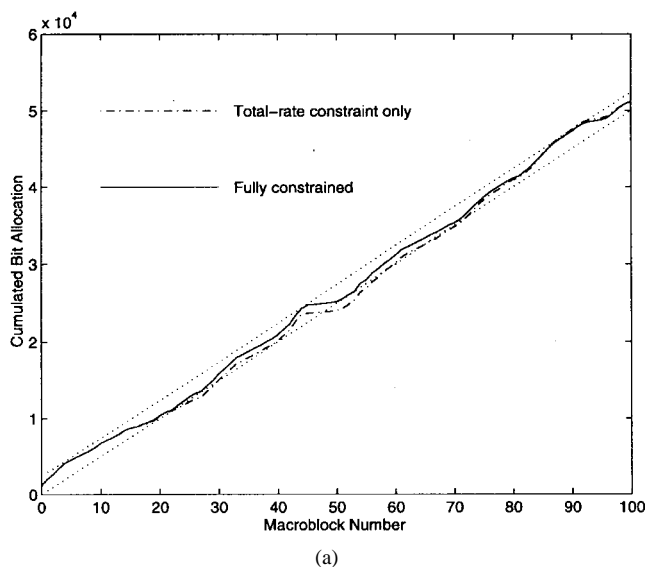


Fig. 7. Bit allocation under optimal independent encoding. Straight lines in each plot mark the upper and lower constraints on cumulated rate at each time. (a) Case of 100 macroblocks. (b) Case of 396 macroblocks. For clarity, curves for (b) are “flattened” by subtracting  $500n$  from the cumulated rate at macroblock  $n$ ; thus, we have the term “normalized.” As a result, the rate constraints actually correspond to encoder buffer boundaries, and the normalized cumulated rates correspond to would-be encoder buffer levels should the buffer be able to accommodate the coded bits.

units under delay encoding. The survivor path with minimum total distortion then gives the solution. With full-tree search, the solution is optimal under either CBR or VBR transmission [9].

The sliding-window video sequence coding procedure can be described as follows. Modification to jumping-window coding is straightforward.

*Algorithm 2:* S1: Grow the coding tree recursively, one level at a time, subject to the rate constraints derived earlier.

S2: Find the path with the least total distortion. Quantize and encode video unit  $n$  using the quantizer step size in the optimal solution. Determine the transmission rate  $t(n)$  subject to (14).

S3: Update the buffer levels  $e(n)$ ,  $r(n)$ , and  $L(n)$ . Increment  $n$  by 1 and go to S1. □



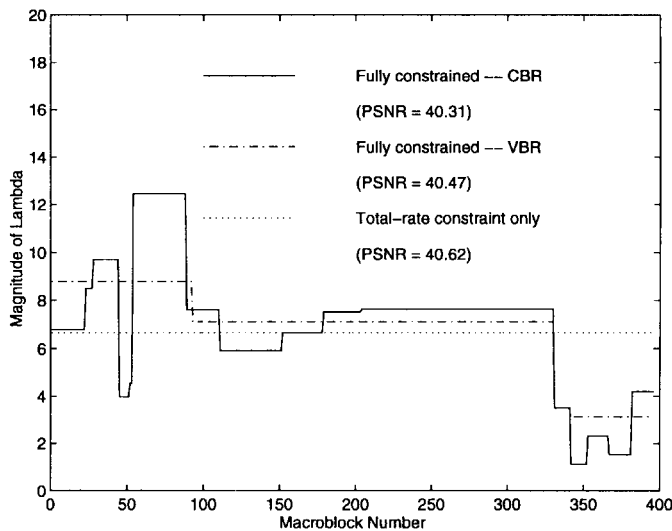


Fig. 8. Variation of optimal Lagrange multipliers with time.

## V. A SIMULATION STUDY

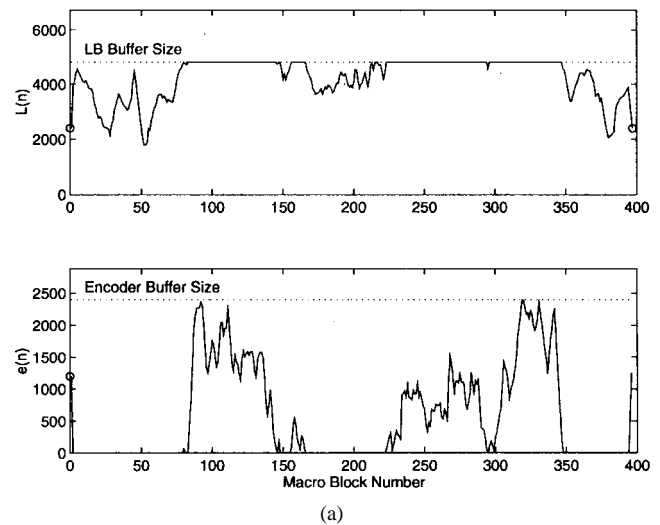
We simulate both independent and dependent coding. The goals are, first, to examine the performance and properties of the (sub)optimal coding of earlier sections, and second, to compare the performance of CBR- and ATM-type VBR transmission. The codec and transmission delay  $d$  is set equal to  $N + 1$  (the delay-encoding “window size”) in all cases. For convenience, we use PSNR to measure coding performance, although it may not be a subjectively meaningful measure at all times.

### A. Independent Coding

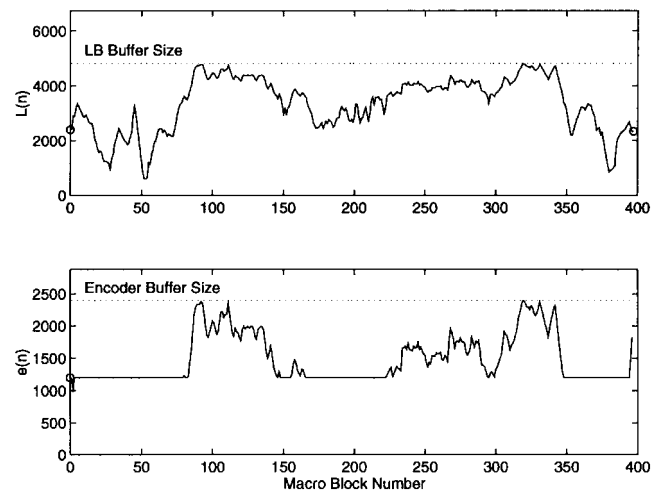
We consider H.261-type intraframe video coding, and let each video unit be a macroblock. The encoder buffer size is set to be 2400 bits, and the channel rate per video unit is 500 bits. We employ jumping-window encoding. The first picture in the well-known salesman sequence, in CIF format (396 macroblocks), is encoded.

We first inspect the variation in video rate from optimized bit allocation. For this, we let  $N = 99$  and encode the first 100 macroblocks of the picture under CBR transmission. The result is plotted in Fig. 7(a). For comparison, the result from optimization only under the total-rate constraint (but not buffer constraints at any earlier time) is also shown. Observe in Fig. 7(a) that, for video units near 20–30 and 45–60, where the buffer constraints are violated in the total-rate-constrained solution, the fully constrained optimal bit allocation is sometimes close to the constraint boundary, as one might expect from the discussions in the Appendix. It is also of interest to observe that the two curves show similar variations over these regions. The rate variation from fully constrained and total-rate-constrained optimization of all 396 macroblocks together is shown in Fig. 7(b).

We next compare the performance and characteristics of CBR and VBR transmission. The LB size and leak rate are set to be 4800 and 500 bits per video unit, respectively. The



(a)



(b)

Fig. 9. LB and encoder buffer fullness under different transmission rate control schemes. (a) High transmission rate. (b) Medium transmission rate.

initial and target LB levels are both set to be half full so that the total bits available for coding are the same under CBR and VBR. Results in Fig. 8 show that VBR offers a slightly higher average PSNR. The difference is small, presumably due to the video material in addition to the use of a suboptimal transmission rate control scheme. We expect a more significant difference in performance for material showing greater variation in complexity across video units. (Indeed, it will be seen later in dependent coding results that VBR yields a prominent gain at scene cuts.) Curves in Fig. 8 also show that, compared to CBR, the Lagrange multipliers under VBR transmission exhibit less variation, and are closer to the total-rate-constrained solution, which implies a smaller deviation of bit numbers from the total-rate-constrained solution, and thus a better match of transmission rate with video complexity.

Reconsider the determination of transmission rates under VBR transmission. Equation (14) practically specifies an infinitude of possible choices. Specifically, we may take the

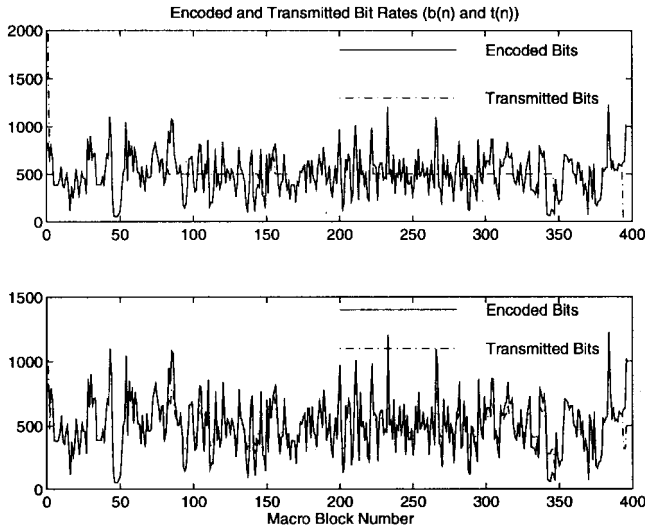


Fig. 10. Encoded and transmitted bit rates. Top: high transmission rate; bottom: medium transmission rate.

weighted sum of the upper and the lower bounds as

$$\sum_{k=n_a}^i t(k) = w_1 \cdot \max\{0, (U_D, U_L, O_E)(n_a, i)\} + w_2 \cdot \min\{(O_D, O_L, U_E)(n_a, i)\} \quad (22)$$

for  $i = n_a, n_a + 1, \dots, n_b$ , where  $w_1, w_2 \geq 0$  and  $w_1 + w_2 = 1$ . Fig. 9(a) is obtained with  $(w_1, w_2) = (0, 1)$ . We see that, in this situation, the encoder buffer is often empty when the LB is not full, which indicates that the LB, or more exactly, the ATM network, is absorbing the variation in the video rate for a best coding quality. Fig. 9(b) is obtained with  $(w_1, w_2) = (\frac{1}{2}, \frac{1}{2})$ , which has led to a half-full encoder buffer when the LB buffer level is not high. As further illustration, the encoded and transmitted bit rates in each case are depicted in Fig. 10, which shows that the transmission rate follows the encoded bit rate when the LB is not full and is kept at the leak rate when it is full. Interestingly, the two transmission rate control methods have resulted in the same received video quality because of the same solution in bit allocation.

We now examine the variation in coding performance as a function of the length of encoding delay. Fig. 11 shows the PSNR results obtained with different coding delays for both CBR and VBR transmission. The best performance occurs at full-length delay. But it is interesting that the PSNR is not monotonically increasing with delay for either CBR or VBR, which can be expected since the forced satisfaction of  $R_b$  for a block of video units may not be optimal over a longer or shorter block. But the suboptimality should lessen as long-enough blocks are used. For VBR transmission, the PSNR improvement over CBR transmission increases as  $N$  increases.

**B. Dependent Coding**

We consider H.261-type coding. Fig. 12 shows the average PSNR results obtained from (sub)optimal CBR and VBR coding of the CIF Salesman sequence at 10 pictures/s at different (average) rates, where the suboptimality arises due to the use of distortion-minimized average quantizer step sizes

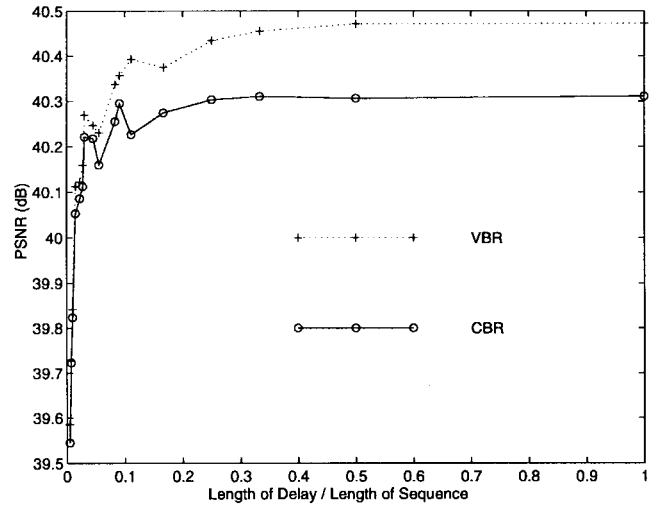


Fig. 11. PSNR performance at different encoding delays (in fractions of total sequence length, i.e., 396 macroblocks).

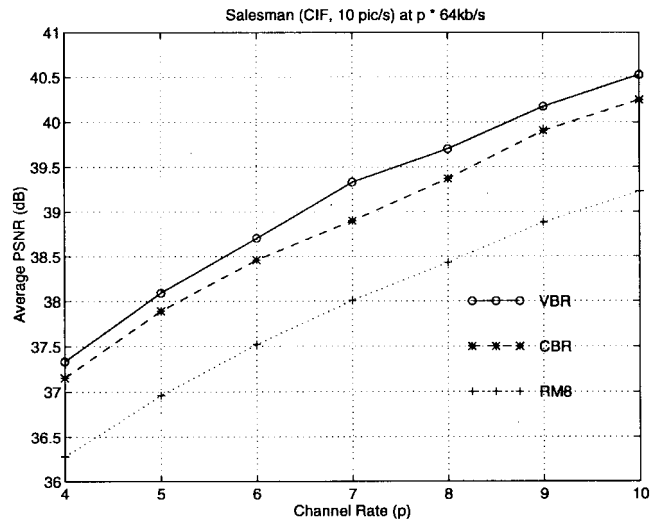


Fig. 12. Overall PSNR of RM8 and delayed coding of CIF Salesman sequence at 10 pictures/s at  $p \times 64$  kbits/s.

for each picture for simplicity. These average quantizer step sizes are limited to between 2 and 36, in steps of two. For comparison, results from the widely used reference algorithm RM8 [18] is also included. For fairness in comparison, for all algorithms, the first picture in the sequence is coded the same as in RM8. The codec buffer sizes are  $E = R = p \times 6400$  where  $p$  is as defined in the figure, and the codec delay  $N$  is two pictures. For VBR transmission, the LB begins (somewhat unfairly) at empty at the second picture. It is seen that the suboptimal CBR coding yields approximately 1.0 dB gain over RM8, and the suboptimal VBR coding gains roughly an additional 0.2 dB over all rates. Fig. 13 shows the variation of per-picture PSNR at an average rate of 384 kbits/s ( $p = 6$ ).

It is of interest to examine the buffer-level variations in the different coding methods. Fig. 14(a) plots the encoder buffer level at the end of each picture for each method. Fig. 14(b) further shows the number of encoded bits and the VBR channel transmission rate. Observe from Fig. 14(a) that the buffer-level variations in the delayed-coding solutions are greater

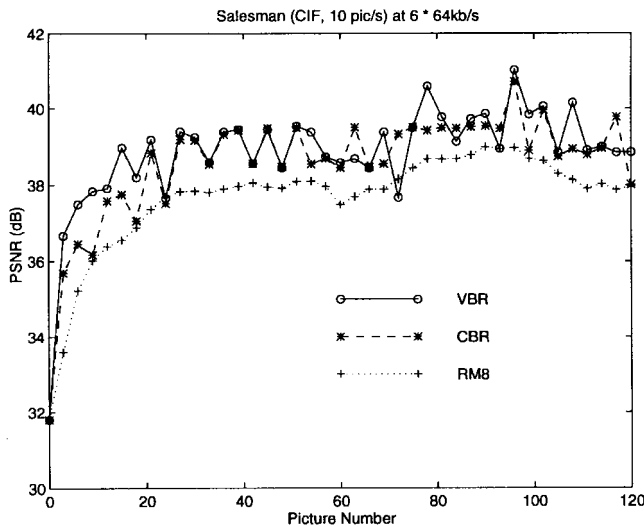


Fig. 13. PSNR of RM8 and delayed coding of CIF Salesman sequence at 10 pictures/s at 384 kbits/s.

than that in RM8. On the positive side, this may be due to the fact that the delayed-coding solution makes better use of the buffer space to catch the complexity variation in successive pictures and adjust the bit rate accordingly. However, there are at least two other possibilities. And it is possible that all of these forces are at work at the same time to different degrees. First, as noted above, for simplicity, we have limited the set of quantizer step sizes, or equivalently, the number of tree nodes at each level. This may have affected the algorithm's ability to find smoothly varying bit allocations for successive pictures. Second, in delayed encoding, if somehow a picture is allocated a high rate, then later pictures may be encoded to a high fidelity more easily (because of a good reference picture in predictive coding), but they also are subject to more stringent rate constraints. Over time, the quality may suffer until a point where the high complexity in a picture (due to a poor reference in motion-compensated prediction) and the gradually released rate constraints again lead to a high-rate coding. The end result may be oscillating rates, PSNR, and buffer levels as shown in Figs. 13 and 14. In VBR coding, where we have an additional degree of freedom in controlling the transmission resource, the above could be further enhanced. However, we should note that oscillation in PSNR, by itself, may not be an undesirable phenomenon since the aim in delayed coding is to achieve minimum *total distortion* over several pictures rather than homogeneous PSNR.

Heuristically, the performance improvement due to delayed coding should be especially acute at a scene cut. This is because of the  $N$ -picture worth of "foreknowledge" provided following the scene cut, while RM8, by its simple feedback control of the quantizer step sizes, does not make use of this foreknowledge, even if available. To verify this point, we consider a composite sequence made up of 30 pictures each of Salesman, Swing, Miss America, and Claire in that order. The composite sequence is named Fly, in which the several scene changes can be used to verify the effectiveness of the delayed-coding schemes in bit allocation and the flexibility offered by VBR ATM rate control.

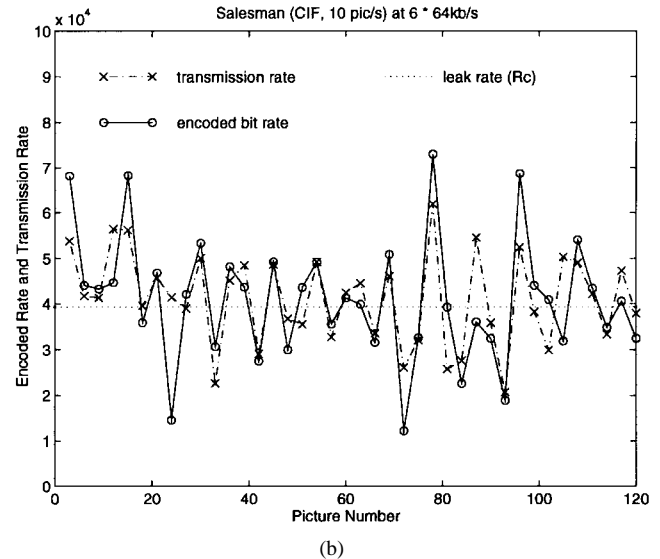
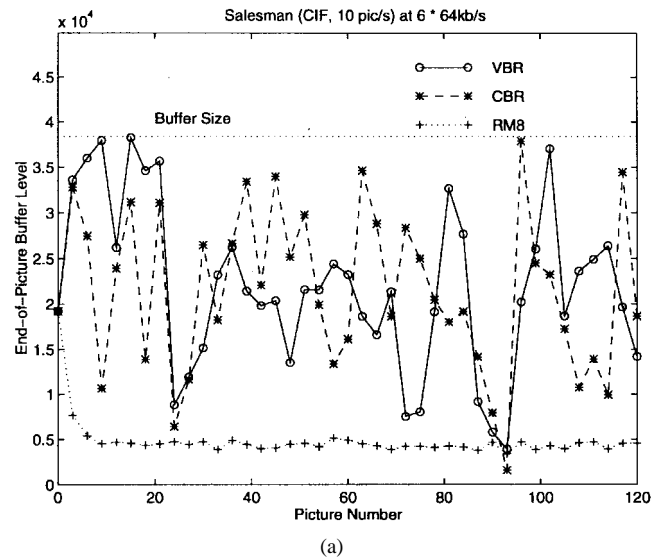
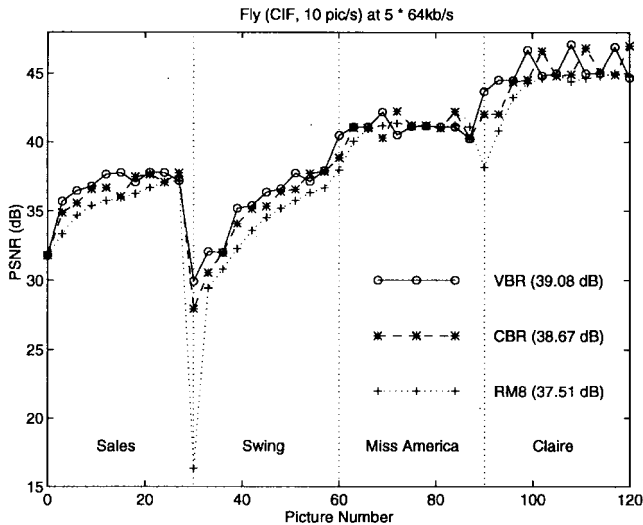


Fig. 14. Variation in buffer level and rate in RM8 and delayed coding of the Salesman sequence. (a) Buffer level variation. (b) Encoded bits and channel transmission rate in VBR coding.

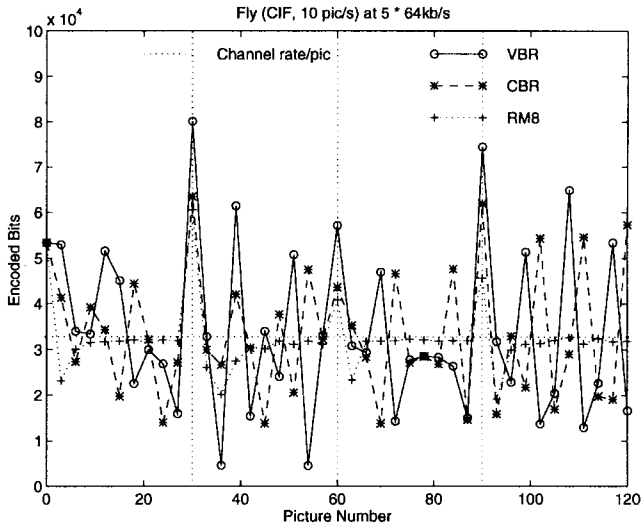
Fig. 15(a) shows the PSNR result from coding Fly at 320 kbits/s. We see that the delayed coding schemes can have a significant advantage immediately following a scene cut, possibly at the expense of a somewhat lower performance immediately preceding it. The PSNR performance of VBR ATM transmission is significantly better than that of CBR at scene changes (although the edge in sequence-wise average PSNR is less significant). And we see from Fig. 15(b) that the VBR ATM transmission, by its greater flexibility in rate control, allocates more bits than CBR or RM8 to the picture after a scene cut.

## VI. CONCLUSION

We considered objective optimal encoding of video sequences for ATM networks. Both independent and dependent coding were investigated. They can be formulated as optimization problems subject to multiple rate constraints arising from the finite channel transmission rate and the finite codec



(a)



(b)

Fig. 15. Coding of the CIF Fly sequence by RM8 and delayed coding at 10 pictures/s at 320 kbits/s. (a) PSNR performance. (b) Bits per picture.

buffer sizes. For independent coding, we derived an efficient bit allocation algorithm based on the use of multiple Lagrange multipliers. In this algorithm, we exploited the monotonic relation between total distortion and the amount of deviation from the constraint boundaries. The algorithm is optimal up to a convex-hull approximation of the D–R relations in the case of CBR transmission. It is suboptimal in the case of VBR transmission due to the use of a suboptimal transmission rate control method. For dependent coding, the aggregate D–R relationship exhibits a tree structure, and the Lagrange multiplier approach becomes rather unwieldy. We resorted to a constrained tree search approach. The solution is optimal for both CBR and VBR transmission if the full constrained tree is searched.

We presented some simulation results which confirm the superiority of the coding quality of the derived encoding algorithms over simpler schemes such as RM8. In addition, (sub)optimal coding under VBR transmission, as expected, has yielded higher PSNR performance than that under CBR trans-

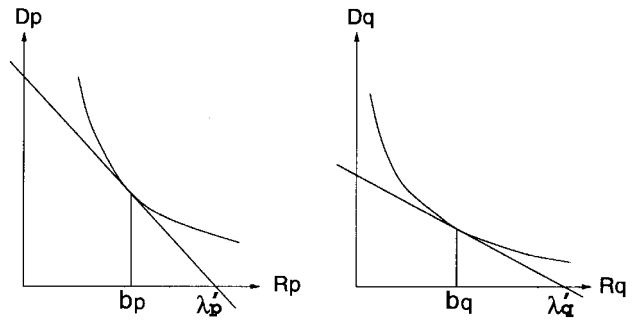


Fig. 16. Nonoptimality with unequal Lagrange multipliers.

mission, although the difference appears not very significant except at scene cuts.

The ability of the optimization approach to efficiently allocate bits among pictures with differing complexity could be further exploited for bandwidth allocation and coding of multiple video sources for transmission over a shared channel.

APPENDIX

THEORY OF THE LAGRANGE-MULTIPLIER CODING ALGORITHM

For simplicity, consider only convex D–R relations. Define  $\lambda'_{n-N+i} \triangleq \sum_{j=i}^N \lambda_j$ ,  $i = 0, 1, \dots, N$ . Then (15) can be rewritten as

$$\begin{aligned} \min_{Q(m), n-N \leq m \leq n} & \left\{ \sum_{m=n-N}^n [D(m) + \lambda'_m b(m+N)] \right\} \\ & = \sum_{m=n-N}^n \min_{Q(m)} [D(m) + \lambda'_m b(m+N)] \end{aligned} \quad (23)$$

where the equality holds due to the independence in the D–R relations. Since there is a one-to-one correspondence between  $\{\lambda_i\}$  and  $\{\lambda'_{n-N+i}\}$ , characterization of  $\{\lambda_i\}$  can be accomplished by characterizing  $\{\lambda'_{n-N+i}\}$ . For convenience, term  $\{\lambda'_{n-N+i}\}$  the *prime Lagrange multipliers*. Now, assume tentatively that the D–R relations associated with the video units are continuous. And assume that the optimal bit allocation touches one of the two boundaries in (9) at time  $v_i$ ,  $i = 1, \dots, V+1$ , where  $v_{V+1} = n$ . In particular, it touches the RHS boundary at  $v_{V+1}$  so that the bit budget  $R_b$  is fully consumed. Further, let  $v_0 = n - N - 1$ . Then we have the following.

*Lemma 1:* The optimal bit allocation is such that  $\lambda'_j$ ,  $j = n-N, \dots, n$ , are constant over each video subsequence  $[v_{i-1} + 1, v_i]$ ,  $i = 1, \dots, V + 1$ .

*Proof:* Suppose, in the optimal solution,  $\lambda'_p < \lambda'_q (< 0)$  for some  $p, q \in [v_{i-1} + 1, v_i]$ . Then in minimization of  $D(m) + \lambda'_m b(m+N)$  for  $m = p, q$ , we have the situation in Fig. 16 where  $b_p$  and  $b_q$  are the optimal solutions. Due to the different slopes in the D–R curves at  $b_p$  and  $b_q$ , we can reduce the total distortion without changing the total rate by moving bits from  $R_q$  to  $R_p$  until either  $\lambda'_p = \lambda'_q$  or until a constraint in (9) is reached somewhere in  $[p, q]$ . The former contradicts the optimality assumption of the original solution, while the latter contradicts the assumption that the boundaries in (9) are not touched in  $(v_{i-1}, v_i)$ .  $\square$

Define the *optimal anchor point*  $a$  as the last video unit before which the optimal prime Lagrange multipliers are equal, i.e.,  $\lambda'_{n-N} = \dots = \lambda'_a \neq \lambda'_{a+1}$ . And call the subsequence  $[n-N, a]$  the *optimal anchor subsequence*. We now prove that the anchor point in each recursion of the algorithm described in Section III will be located at or after the optimal anchor point. The proof consists of two steps. First, we show that the anchor point  $v$  obtained from the initial optimization (yielding optimal  $\lambda_N$ ) is located at or after the optimal anchor point  $a$ . Then we show that the anchor point will stay at or after the optimal anchor point with each recursion in the algorithm.

*Lemma 2:* The initial anchor point  $v$  obtained by optimization subject only to the total rate constraint  $R_b$  is such that  $v \geq a$ .

*Proof:* We showed earlier that the sequence of optimal prime Lagrange multipliers  $\{\lambda'_j\}$  is piecewise constant with magnitude changes occurring only at some  $v_i, i \in \{1, \dots, k\}$ . Four cases can be envisioned: 1) wherever  $\lambda'_{v_i} \neq \lambda'_{v_i+1}$  ( $i = 1, \dots, k$ ), the boundary in (9) that is touched at  $v_i$  is the RHS boundary; 2) the RHS boundary in (9) is touched at  $a$ , and there exists some time  $j > a$  where the LHS boundary is touched and  $\lambda'_j \neq \lambda'_{j+1}$ ; 3) same as 1), but replace the RHS by the LHS; and 4) same as 2), but replace the RHS by the LHS and vice versa. We address the first two cases only since the other two are complementary.

For case 1), we show that  $\lambda'_{v_i} \leq \lambda'_{v_i+1} \forall i$  and  $\lambda'_a < \lambda_N$ , and the conclusion follows. To show that  $\lambda'_{v_i} \leq \lambda'_{v_i+1}$ , assume  $\lambda'_{v_i} > \lambda'_{v_i+1}$ . Then we have the situation of Fig. 16, with  $\lambda'_{v_i}$  and  $\lambda'_{v_i+1}$  in the roles of  $\lambda'_q$  and  $\lambda'_p$ , respectively. Bits can be moved from video unit  $v_i$  to  $v_i + 1$  to reduce distortion while keeping the same total rate. But then the total rate at  $v_i$  would shift inside the RHS boundary in (9), contradicting the assumption that the optimal solution touches that boundary there. To show that  $\lambda'_a < \lambda_N$ , assume  $\lambda_N \leq \lambda'_a$ . Then since  $\lambda'_j$  is nondecreasing in  $j$ ,  $\lambda_N \leq \lambda'_j \forall j$  and  $\lambda_N < \lambda'_j \forall j > a$ . By convexity of the D-R relations, the total rate obtained under  $\lambda_N$  would be lower than in the optimal allocation, contradicting the assumption that  $\lambda_N$  satisfies the total rate constraint. Therefore,  $\lambda_N > \lambda'_a$ . By convexity in the D-R relations, bits allocated to video units  $[n-N, a]$  under  $\lambda_N$  are more than that under  $\lambda'_a$ . Hence, there is overflow at  $a$  and no underflow before it, and thus  $v \geq a$ .

For case 2), we can show that  $\lambda'_a < \lambda'_{a+1}$ . Then if  $\lambda_N \leq \lambda'_a$ , there is no overflow up to  $a$  and there is underflow afterwards; if  $\lambda_N > \lambda'_a$ , then there is overflow at  $a$  and no underflow before it. The proof relies on D-R convexity as in case 1). In either case,  $v \geq a$ .  $\square$

*Lemma 3:* The anchor point  $v$  will stay at or after the optimal anchor point with each recursion in the algorithm.

*Proof:* Consider the two cases addressed in the proof of the last lemma. For case 1), we begin the recursion by having overflow at some anchor  $v \geq a$  which resulted from optimization with some Lagrange multiplier  $\lambda_f$  geared to satisfy a rate constraint at some "future" video unit  $v_f > v$ . Optimization of coding for video units  $[n-N, v]$  subject only to the RHS constraint of (9) at video unit  $v$  by using a Lagrange multiplier  $\lambda_v$  will lead to a solution characterized by some  $\lambda_v > \lambda'_a$  in case  $v > a$  and  $\lambda_v = \lambda'_a$  in case  $v = a$ . The proof

involves an examination of the convexity in D-R relations and the nondecreasing nature of  $\lambda'_j$  as in the last lemma; details are omitted. Now, if  $\lambda_v = \lambda'_a$ , then we have obtained the optimal anchor point at  $v = a$  together with the optimal coding of  $[n-N, a]$ ; and we can proceed to the optimal coding of  $[a+1, n]$ . If  $\lambda_v > \lambda'_a$ , then we are in a similar situation as in the proof for case 1) in the last lemma where we had  $\lambda_N > \lambda'_a$ , and thus a similar conclusion holds. For case 2), we begin with either an underflow at  $v > a$  or an overflow at  $v \geq a$ . In case of underflow, optimal coding of  $[n-N, v]$  subject to the LHS constraint in (9) at  $v$  either will result in an overflow at  $a$  with no underflow before it, or will result in an underflow at a later video unit with no overflow before that video unit. The proof again involves a look into the D-R convexity. In case of overflow, the situation is similar either to the immediately preceding underflow case or to case 1) discussed above, depending on where  $v$  is located.  $\square$

Lemmas 2 and 3 together establish the convergence of the Lagrange-multiplier algorithm toward the optimal solution in independence coding for continuous D-R relations. The discussion can be extended to handle discrete D-R relations [15].

The situation with dependent coding is much more complicated because the optimization objective (15) can no longer be decomposed as in the RHS of (23). For example, in the  $N = 1$  case depicted in Fig. 3 with the initial optimal solution under  $\lambda_N$  located at  $A$ , the anchor subsequence  $[n-1, n-1]$  cannot be optimized independently of  $[n, n]$ . In other words, the Lagrange multiplier  $\lambda_0$  cannot be optimized independently of  $\lambda_1$ . The algorithm in Section III can be modified to accommodate this dependence, but the resulting computation can become overwhelming.

#### ACKNOWLEDGMENT

The authors would like to thank the reviewers for their comments which helped improve the paper.

#### REFERENCES

- [1] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
- [2] J.-J. Chen and H.-M. Hang, "A transform video coder source model and its application," in *Proc. IEEE Int. Conf. Image Processing*, vol. II, Nov. 1994, pp. 967-971.
- [3] Y. Takishima, M. Wada, and H. Murakami, "An analysis of optimal frame rate in low bit rate video coding," *IEICE Trans. Commun.*, vol. E76-B, pp. 1389-1397, Nov. 1993.
- [4] *Video Codec for Audiovisual Services at P × 64 kbit/s*, ITU-T Recommendation H.261.
- [5] *Information Technology—Generic Coding of Moving Pictures and Associated Audio: Video*, ISO/IEC 13818-2 and ITU-T Recommendation H.262.
- [6] *Video Coding for Low Bitrate Communication*, Draft ITU-T Recommendation H.263, Oct. 1995.
- [7] *Coding of Moving Pictures and Associated Audio—for Digital Storage Media at Up to About 1.5 Mbit/s—Part 2: Video*, ISO/IEC 11172-2.
- [8] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445-1453, Sept. 1988.
- [9] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximation," *IEEE Trans. Image Processing*, vol. 3, pp. 26-40, Jan. 1994.
- [10] C.-Y. Hsu, A. Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," this issue, pp. 1016-1028.

- [11] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multi-resolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.
- [12] D. W. Lin, M.-H. Wang, and J.-J. Chen, "Optimal delayed-coding of video sequences subject to a buffer-size constraint," *SPIE, vol. 2094, Visual Commun. Image Processing*, pt. 1, pp. 223–234, 1993.
- [13] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [14] E. P. Rathgeb, "Policing of realistic VBR video traffic in an ATM network," *Int. J. Digital Analog Commun. Syst.*, vol. 6, pp. 213–226, 1993.
- [15] D. W. Lin and J.-J. Chen, "Efficient optimal rate-distortion coding of video sequences under multiple rate constraints," in *Proc. IEEE Int. Conf. Image Processing*, 1997.
- [16] J.-J. Chen and D. W. Lin, "Optimal video coding over ATM networks," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 1995, pp. 21–24.
- [17] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 361–372, Dec. 1992.
- [18] *Description of Ref. Model 8 (RM8)*, Doc. 525, CCITT SG XV, Working Party 4, Specialist Group on Coding for Visual Telephony, June 9, 1989.



**Jiann-Jone Chen** was born in Taichong, Taiwan, R.O.C., in 1966. He received the B.S. and M.S. degrees from National Cheng Kung University, Tainan, Taiwan, in 1989 and 1991, respectively, and the Ph.D. degree from National Chiao Tung University, Hsinchu, Taiwan in 1997, all in electrical engineering.

He is currently working toward the Ph.D. degree in electronics engineering at National Chiao Tung University. His research interests include digital image processing and digital video coding.



**David W. Lin** (S'79–M'81–SM'88) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, R.O.C, in 1975, and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1979 and 1981, respectively, all in electrical engineering.

He was with Bell Laboratories during 1981–1983, and with Bellcore during 1984–1994. Since 1990, he has been a Professor in the Department of Electronics Engineering and the Center for Telecommunications Research, National Chiao Tung University. He has conducted research in the areas of digital adaptive filtering and telephone echo cancellation, digital subscriber line and coaxial network transmission, speech coding, and video coding. His research interests include various topics in signal processing and communication engineering.