Efficient bit allocation under multiple constraints on cumulated rates for delayed video coding

David W. Lin* and Jiann-Jone Chen

Dept. of Electronics Engineering and Center for Telecommunications Research
National Chiao Tung University
Hsinchu, Taiwan 300, ROC
*Email: dwlin@cc.nctu.edu.tw

ABSTRACT

We consider optimal encoding of a sequence of video units under a given set of rate constraints which may arise from finite codec delay, finite channel capacity, and finite codec buffer sizes. A Lagrange-multiplier approach is employed and some useful properties of the optimal Lagrange-multiplier solution are obtained under the assumption that the allowed video data rates are continuous. Based on these properties, we derive two solution algorithms for discrete rate allocation. The algorithms are more efficient than that have been presented to date. The solution is optimal when the distortion-rate relations of the video units are convex and the selectable rates of the video units are uniformly spaced with the same granularity. When these conditions do not hold, the Lagrange-multiplier solution may be suboptimal, but can be improved or optimized by a search about the solution.

Keywords: video sequence coding, bit allocation, quantizer control, buffer control, Lagrange-multiplier optimization

1 INTRODUCTION

In video coding, the encoder has the task of producing highest-possible coded video quality subject to a set of constraints on codec delay, codec buffer sizes and channel transmission rate.[1,2] In packet networks such as ATM, the channel buffer size or policing mechanism also enters the constraining relationship.[4,5,2] In typical video coding schemes such as ITU-T's H.26x[6–8] and ISO's JPEG[9] and MPEGx,[10,7] the coded video quality is controlled by choice of quantizer scales for, or equivalently, allocation of available bits to, the video units in the video sequence. An efficient algorithm for optimal bit allocation (that minimizes a sum-distortion) under a single total-rate constraint has been derived by Shoham and Gersho.[11] The algorithm employs a Lagrange-multiplier optimization technique. A few researchers, including the present authors, have also attacked the more complicated problem of optimal bit allocation (minimizing a sum-distortion) under multiple constraints for delayed video coding.

A tree/trellis-search approach is discussed by some.[12–15,2] The approach is based on the observation that, beginning at a certain point, the collection of all possible quantizer choices for all subsequent video units forms a tree. Hence the desired optimal bit allocation can be obtained by a search over the tree. For "independent coding" where successive video units possess independent D-R (distortion-rate) relations, the algorithm can be simplified by observing that each tree path defines a sequence of cumulated coded data rates, or equivalently, a sequence of codec buffer levels. The collection of all allowed rates or buffer levels at each time can be treated as states and the coding tree can thus be arranged into a trellis. A Viterbi-type algorithm can then be employed to obtain the optimal

SPIE Vol. 3024 • 0277-786X/97/$10.00

constrained bit allocation over the tree/trellis. However, the computational load quickly becomes overwhelming with the size of the tree/trellis.

The Lagrange-multiplier approach provides an efficient solution in this case, though probably suboptimal due to its confinement of the solution to the convex hull of certain D-R relations.[1,2,16] The approach has been taken to investigate optimal delayed video coding for constant bit-rate transmission[1] as well as variable bit-rate transmission.[2,3] Different solution algorithms have been presented.[16,2,3] One algorithm[16] only controls buffer overflows and is optimal in a certain sense (to be further characterized later) when only buffer overflows may occur. When buffer underflows may occur, it ceases to be optimal. Another algorithm[2,3] controls both buffer over- and underflows. However, the complexity of the algorithm is seen to show a difference, between the worst case and some more favorable cases, of one to two orders of magnitude in the number of video units in the delayed-coding window. Moreover, it is desirable to have more in-depth characterization of the discrete bit allocation solution. In this paper, we present more efficient algorithms based on the Lagrange-multiplier approach. And we comment on the solution's properties in discrete rate allocation.

In what follows, Sec. 2 formulates the optimization problem and introduces the Lagrange-multiplier approach. Sec. 3 discusses the properties of the optimal Lagrange-multiplier solution which are of use in arriving at efficient solution algorithms, under the assumption that the allowed video data rates are continuous. Based on these properties, Sec. 4 derives two solution algorithms for discrete rate allocation and considers their complexity. It also discusses the properties of the algorithm solutions in discrete rate allocation. Finally, Sec. 5 gives the conclusion.

2 PROBLEM FORMULATION

Consider delayed coding with delay equal to N video units, where a video unit may be any pertinent grouping of the picture elements, such as (in MPEG terms) a picture, a slice, or a macroblock. That is, at some time n the encoder buffers up the most recent $N+1$ video units (i.e., video units $n-N$ through n) and conducts a joint bit allocation for these video units together. Such delayed source coding is long known to be able to yield better performance than non-delayed coding.[17] Several variants of the same delayed-coding theme can be envisioned. For example, we may consider *sliding-window coding* in which only the video unit $n-N$ is actually encoded at time n (although video units $[n-N+1, n]$ are employed in optimizing the bit allocation), then video unit $n-N+1$ at time $n+1$, etc. For another example, we may consider *jumping-window coding* in which not only bit allocation but also actual encoding is done for video units $[n-N, n]$ at time n, then the encoder is halted until time $n+N+1$ when bit allocation and encoding of video units $[n+1, n+N+1]$ are done, etc.

Let $b(n-N)$ be the number of bits allocated to video unit $n-N$ by the encoder at time n. Normally, $b(\cdot)$ are subject to constraints of the following form

$$L(n-N,k) \le \sum_{i=n-N}^{k} b(i) \le U(n-N,k), \quad k = n-N, n-N+1, \cdots, n, \tag{1}$$

where $L(n-N,k)$ and $U(n-N,k)$ are bounds arising from constraints on codec delay, codec buffer sizes and channel transmission rate.[1,2] Violation of the lhs constraints usually corresponds to encoder buffer underflow and violation of the rhs constraints usually corresponds to encoder buffer overflow. Hence these violations will sometimes be referred to as underflow and overflow, respectively.

Consider a sum-of-distortion performance measure. Then the optimization objective is

$$\min_{Q(i), n-N \le i \le n} \sum_{i=n-N}^{n} D(i) \tag{2}$$

subject to the above constraints, where $Q(i)$ denotes all possible ways of coding video unit i and $D(i)$ denotes the

corresponding distortion in this video unit. Applying the Lagrange multiplier method, we obtain

$$\min_{Q(i),n-N\le i\le n}\{\sum_{i=n-N}^{n} D(i)+\lambda_0 b(n-N)+\lambda_1\sum_{i=n-N}^{n-N+1} b(i)+\cdots+\lambda_N\sum_{i=n-N}^{n} b(i)\}\triangleq J, \tag{3}$$

where λ_i, $i=0,1,\cdots,N$, are the Lagrange multipliers. The desired optimal solution can be obtained by carrying out the above minimization with a proper set of Lagrange multiplier values such that the constraints (1) are satisfied. The key to obtaining the solution is therefore finding this optimal set of Lagrange multiplier values.

Before proceeding, we note that 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 non-convex.[11] This issue will be further touched on later. For the time being, assume that the D-R relations possess the required convexity for the Lagrange multiplier solution to be optimal.

For convenience, define the *prime Lagrange multipliers*

$$\lambda'_{n-N+i}\triangleq\sum_{j=i}^{N}\lambda_j \tag{4}$$

where $i=0,1,\cdots,N$, so that

$$J=\min_{Q(i),n-N\le i\le n}\{\sum_{i=n-N}^{n}[D(i)+\lambda'_i b(i)]\}=\sum_{i=n-N}^{n}\min_{Q(i)}[D(i)+\lambda'_i b(i)]. \tag{5}$$

In writing the last equality, we have assumed independence among D-R relations of the video units $[n-N,n]$. This is the case, for example, for motion JPEG coding and certain ways of intraframe coding. However, even if the D-R relations exhibit dependency, the ensuing algorithms can still be employed to effect a suboptimal solution.

There is a one-to-one correspondence between $\{\lambda_i\}$ and $\{\lambda'_{n-N+i}\}$. Hence the characterization of the optimal Lagrange multipliers can be accomplished by characterizing the optimal prime Lagrange multipliers. Since Lagrange multipliers define slopes on the constituent functions in an optimization, for monotone decreasing D-R functions (for the video units) the prime Lagrange multipliers are nonpositive.[18]

To proceed, assume tentatively that the D-R relations associated with the video units are continuous and strictly convex. Solution algorithms are derived under this assumption and applied to discrete rate allocation. Some characteristics of the ensuing discrete solution are then discussed.

3 PROPERTIES OF THE OPTIMAL SOLUTION

Assume the optimal bit allocation touches one of the two boundaries in (1) at the end of video units v_i (i.e., for $k=v_i$), where $i=1,\cdots,V+1$ and $v_i\in[n-N,n]$ $\forall i$. In particular, it touches the upper boundary at the end of video unit n with $n=v_{V+1}$ so as to fully utilize the available transmission capacity. This last assumption can be interpreted as letting

$$L(n-N,n)=U(n-N,n). \tag{6}$$

Further, let $v_0=n-N-1$. Then we have the following result.

LEMMA 3.1. **(The Segmental Uni-Slope Property)** *The optimal bit allocation is such that the prime Lagrange multipliers λ'_j, $j=n-N,\cdots,n$, are constant over each video subsequence $[v_{i-1}+1,v_i]$, $i=1,\cdots,V+1$.*

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 $D(j)+\lambda'_j b(j)$ for $j=p,q$ we have the situation in Fig. 1 where b_p and b_q are the optimal solutions. Due to the

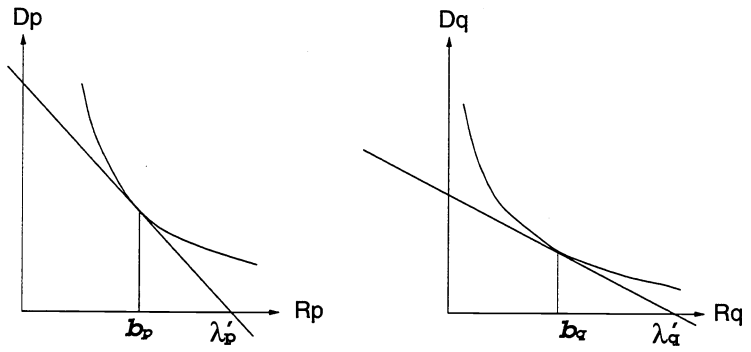

Figure 1: Nonoptimality with unequal Lagrange multipliers in the case of continuous D-R functions.

different slopes in 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 (1) 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 (1) are not touched in (v_{i-1}, v_i). □

Define the *docker points*, or simply *dockers*, d_j $(j = 1, 2, \cdots)$ as those v_i across which the optimal prime Lagrange multipliers change values. For example, the first docker point d_1 is the last video unit before which the optimal prime Lagrange multipliers are equal, i.e., $\lambda'_{n-N} = \cdots = \lambda'_{d_1} \neq \lambda'_{d_1+1}$. Assume there are A such points. For convenience, define two additional dockers $d_0 = n - N - 1$ and $d_{A+1} = n$. Fig. 2 illustrates the concept of docker points. Term a docker at which the rate allocation touches the lower (resp. upper) boundary in (1) a *lower (resp. upper) docker*. And term the subsequence $(d_{j-1}, d_j]$ the jth *docker subsequence*. Since $L(n - N, n) = U(n - N, n)$, d_{A+1} is both a lower docker and an upper docker.

The significance of the docker points is that, once we know where they are, the total rate for each docker subsequence, given by $\sum_{i=d_{j-1}+1}^{d_j} b(i)$ for $j = 1, \cdots, A$, can be determined from (1). And the problem of optimal bit allocation subject to the multiple constraints in (1) is simplified to A independent problems, one for each subsequence subject to only one rate constraint. The latter can be solved, for example, using Shoham and Gersho's technique.[11] The following property of the optimal prime Lagrange multipliers is of use in arriving at a method to find the dockers.

LEMMA 3.2. **(The Slope-Change Property)** *The optimal prime Lagrange multipliers are such that* $\lambda'_{d_i} < \lambda'_{d_i+1}$ *for any upper docker point* d_i *and* $\lambda'_{d_i} > \lambda'_{d_i+1}$ *for any lower docker point* d_i*, where* $i \in [1, A]$.

Proof. The proof relies on a look into the convexity of D-R relations as in the previous lemma. Consider only the case where d_i is an upper docker, as the other case is complementary. Suppose the optimal prime Lagrange multipliers are such that $\lambda'_{d_i} > \lambda'_{d_i+1}$. Then for the optimal solution we have the situation depicted in Fig. 1 with d_i in the role of q and $d_i + 1$ in the role of p. Due to the different slopes in 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, i.e., from video unit d_i to video unit $d_i + 1$. But this would shift the total rate up to d_i inside the upper boundary specified in (1) and thus contradict the assumption that the optimal solution touches that boundary there. □

Our objectives are an efficient way to determine if dockers exist in $[n - N, n)$ and, in their presence, an efficient way to locate them. For the first objective, consider the minimization

$$\sum_{i=n-N}^{n} \min_{Q(i)} [D(i) + \lambda' b(i)] \tag{7}$$

where λ' is some number, given or to be determined according to some criterion. Note that this minimization is

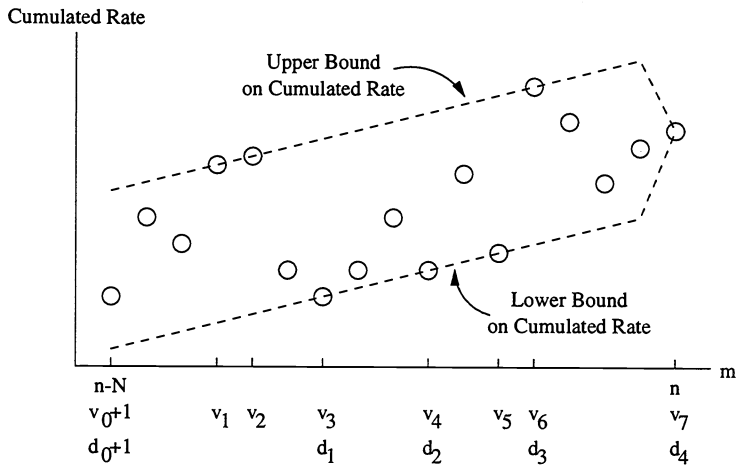

Figure 2: Concept of docker points. Abscissa denotes time (index of video units). Ordinate positions of circles denote optimal cumulated bit allocation for video units $[n-N, m]$. The plot assumes that $\lambda'_{v_1} = \lambda'_{v_2} = \lambda'_{v_3} \neq \lambda'_{v_4} \neq \lambda'_{v_5} = \lambda'_{v_6} \neq \lambda'_{v_7}$. For simplicity, the upper and the lower bounds on cumulated rate are drawn as straight lines, which is the case for constant bit-rate transmission. For variable bit-rate transmission the bounds may be jagged.

equivalent to (3) with $\lambda_0 = \cdots = \lambda_{N-1} = 0$ and $\lambda_N = \lambda'$. It is also equivalent to (5) with $\lambda'_{n-N} = \cdots = \lambda'_n = \lambda'$. Now consider the minimization (7) subject to the total-rate constraint, i.e.,

$$\sum_{i=n-N}^{n} b(i) = U(n-N, n), \tag{8}$$

by proper choice of the λ'. For convenience, term this optimization a *trial optimization* over $[n-N, n]$, the ensuing solution a *trial solution*, and the resulting value of λ' the associated *optimal trial Lagrange multiplier*. We have the following result which can be employed to check whether a docker exists. The proof is given in the full paper.[19]

THEOREM 3.3. **(The Docker-Free Condition)** *There exists no docker point in $[n-N, n)$ if and only if the trial solution for $[n-N, n]$ satisfies all the rate constraints specified in (1). Further, when either proposition holds, the trial solution gives the desired constrained optimal solution to (2).*

Suppose there is at least one docker in $[n-N, n)$, i.e., $A > 0$. It appears difficult to further characterize an arbitrary docker to the extent that they can be easily identified, except the first and the last in $[n-N, n)$. We only discuss the properties of the first docker. Properties of the last docker are complementary.

Two cases can be differentiated: (i) the first docker in $[n-N, n)$ is an upper docker; and (ii) the first docker in $[n-N, n)$ is a lower docker. Consider Case (i) first. Let d_K be the first lower docker $[n-N, n]$. (Recall that $d_{A+1} = n$ is both an upper and a lower docker. Hence a lower docker always exists in $[n-N, n]$.) Then by the Slope-Change Property, $\lambda'_{d_1} < \lambda'_i$ for $d_1 < i \leq d_K$. Consider (7). If $\lambda' = \lambda'_{d_1}$, then by convexity of the D-R relations, the $b(i)$ obtained from this minimization will be smaller than that in the optimal solution, for $d_1 < i \leq d_K$. As a result, underflow will occur at d_K (and possibly even before it) while no overflow may occur up to d_K. Situation for Case (ii) is complementary to that for Case (i). Further derivation yields the following result.[19]

THEOREM 3.4. **(The First-Docker Identification Theorem)** *There exists at least a docker point in $[n-N, n)$ and the first of which is an upper (resp. lower) docker if and only if there exists a λ' which yields a solution of the following properties to (7): (i) there exists some $p \geq n-N$ such that the rate constraints (1) are satisfied for video units $[n-N, p]$ (i.e., for $k = n-N, \cdots, p$); (ii) the upper (resp. lower) rate boundary is touched at p; and (iii)*

there exists some $q > p$ such that the rate constraints (1) are satisfied for video units (p, q), with strict inequality for the rhs (resp. lhs) constraints, and such that the bit allocation commits underflow (resp. overflow) at video unit q. Further, when the second proposition holds, we have $\lambda' = \lambda'_{d_1}$ and $p = d_1$.

Based on the fact the first docker is relatively easily identifiable, we consider an approach to the optimal bit allocation problem by way of successively identifying the dockers from the first to the last. We first derive a prototype algorithm which employs binary search to find the dockers. We analyze the algorithm complexity. Then we present ways to accelerate the algorithm, one of which uses the results in Chen and Lin.[2]

4 EFFICIENT BIT ALLOCATION ALGORITHMS

4.1 The basic forward progression algorithm

To effect an efficient solution, we consider binary search for the first docker. The search proceeds as follows. First, a trial optimization over $[n - N, n]$ is conducted. If some rate constraints are violated in the trial solution, then by the Docker-Free Condition we know that at least one docker exists. To search for the first docker, let $n_1 = n - N/2$ and conduct the minimization (7) subject to some rate constraints over the video subsequence $[n - N, n_1]$. The constraints should be chosen so as to facilitate determination of whether the first docker is located before, at, or after n_1. We find that the following two sets of constraints can be used, viz.,

$$\sum_{i=n-N}^{k} b(i) \leq U(n - N, k) \quad \forall k \in [n - N, n_1], \tag{9}$$

with exact equality at some k, and

$$\sum_{i=n-N}^{k} b(i) \geq L(n - N, k) \quad \forall k \in [n - N, n_1], \tag{10}$$

also with exact equality at some k. For convenience, the minimization (7) subject to either (9) or (10) is termed a *docked trial optimization*, the associated minimization solution a *docked trial solution*, and the associated optimal Lagrange multiplier value an *optimal docked trial Lagrange multiplier*. When it is of help to distinguish between the two cases, the former is further modified by *upper* (e.g., upper docked trial optimization) and the latter by *lower*. Let λ'_u and λ'_ℓ denote the optimal upper and lower docked trial Lagrange multiplier values, respectively.

Suppose now we have determined whether the first docker is located before or after n_1. Then the value of n_1 is increased or decreased by $N/4$ accordingly, as one would in a binary search, and two new docked trial optimizations are conducted over the new subsequence $[n - N, n_1]$. This continues, with halved increment/decrement to n_1 with each iteration, until we find d_1, the first docker. Then the subsequence $(d_1, n]$ takes over the role of the original sequence $[n - N, n]$ and the whole procedure is repeated on it to find the first docker therein.

To see how the relation between d_1 and n_1 can be determined from the docked trial solutions, we examine the properties of these solutions under the three possible conditions: (i) $n_1 = d_1$, (ii) $n_1 < d_1$, and (iii) $n_1 > d_1$. Condition (iii) can be further divided into (a) situation where the dockers located in $[n - N, n_1]$ are of the same kind (upper or lower) and (b) situation where both kinds of dockers exist in $[n - N, n_1]$. The condition $n_1 = d_1$ can be identified by checking the trial solutions against the properties listed in the First-Docker Identification Theorem. For the other conditions, we have the following result, whose proof is given in the full paper.[19]

THEOREM 4.1. **(Properties of the Docked Trial Solutions)** *In Condition (ii) (i.e., that $n_1 < d_1$), we have $\lambda'_u \geq \lambda'_{d_1}$ and $\lambda'_\ell \leq \lambda'_{d_1}$. In case $\lambda'_u > \lambda'_{d_1}$, the upper docked trial solution will commit overflow at some video unit $u > n_1$ while satisfying all the rate constraints in (1) for video units $[n - N, u)$ (i.e., for $k = n - N, \cdots, u$ in (1)). Conversely, in case $\lambda'_\ell < \lambda'_{d_1}$, the lower docked trial solution will commit underflow at some video unit $\ell > n_1$ while satisfying all the rate constraints in (1) for video units $[n - N, \ell)$.*

In Condition (iii)(a) (i.e., that $n_1 > d_1$ and the dockers located in $[n-N, n_1]$ are of the same kind), we have either $\lambda'_u = \lambda'_{d_1}$ or $\lambda'_\ell = \lambda'_{d_1}$. And the properties given in the First-Docker Identification Theorem can be checked to determine which case holds.

In Condition (iii)(b) (i.e., that $n_1 > d_1$ and both kinds of dockers exist in $[n-N, n_1]$), it is also possible that $\lambda'_u = \lambda'_{d_1}$ or $\lambda'_\ell = \lambda'_{d_1}$. Further, we have $\lambda'_u < \lambda'_\ell$. And the upper docked trial solutions will commit underflow somewhere in $[n-N, n_1]$ and the lower docked trial solution will commit overflow somewhere in $[n-N, n_1]$.

Based on the above theorem, a method to determine the relation between d_1 and n_1 can be easily derived. Note that in Conditions (i) and (iii)(a), we must have $\lambda'_u = \lambda'_{d_1}$ or $\lambda'_\ell = \lambda'_{d_1}$, which can be identified by checking the properties listed in the First-Docker Identification Theorem. In Conditions (ii) and (iii)(b), it is also possible that $\lambda'_u = \lambda'_{d_1}$ or $\lambda'_\ell = \lambda'_{d_1}$. When these equalities do not hold, Conditions (ii) and (iii)(b) are differentiated by whether $\lambda'_u > \lambda'_\ell$ or $\lambda'_u < \lambda'_\ell$.

To arrive at a complete solution algorithm for the constrained optimization problem (2), note that it is for notational convenience that the preceding theorems are couched in terms of optimization over $[n-N, n]$. They can be made to apply to any subsequence of $[n-N, n]$, say $[n_\ell, n_u]$, by simply redefining $n_u = n$ and $n_u - n_\ell = N$, as long as the beginning and ending rate sums for this subsequence, i.e., $\sum_{i=n-N}^{n_\ell - 1} b(i)$ and $\sum_{i=n-N}^{n_u} b(i)$, are known. In particular, the theorems can be made to apply to the subsequence $(d_i, n]$ by redefining $N = n - d_i - 1$, provided we know whether d_i is an upper or a lower docker. Of course, in this case the d_1 in these theorems would correspond to d_{i+1} in the original problem. We thus obtain the following algorithm.

ALGORITHM 1. **(The Basic Forward Progression Algorithm)**

S0 (Initialization) *Let $n_0 = n - N$, $\Delta = N$, and $i = 1$.*

S1 (Trial optimization, optional) *Do trial optimization over $[n_0, n_0 + \Delta]$. If all the rate constraints associated with video units $[n_0, n_0 + \Delta]$ are satisfied (i.e., if (1) holds $\forall k \in [n_0, n_0 + \Delta]$), then exit.*

S2 *Let $\Delta = \Delta/2$ and $n_1 = n_0 + \Delta$.*

S3 (Upper docked trial optimization) *Do upper docked trial optimization over $[n_0, n_1]$. Check if there exist $p, q \in [n_0, n)$ that satisfy Properties (i), (ii), and (iii) in the First-Docker Identification Theorem. If so, then we have $d_i = p$ and $\lambda'_{d_i} = \lambda'_u$; let $n_0 = d_i + 1$, $\Delta = n - n_0$, and $i = i + 1$ and go to* **S1**. *If not, then go to the next step.*

S4 (Lower docked trial optimization) *Do lower docked trial optimization over $[n_0, n_1]$. Check if there exist $p, q \in [n_0, n)$ that satisfy Properties (i), (ii), and (iii) in the First-Docker Identification Theorem. If so, then we have $d_i = p$ and $\lambda'_{d_i} = \lambda'_\ell$; let $n_0 = d_i + 1$, $\Delta = n - n_0$, and $i = i + 1$ and go to* **S1**. *If not, then go to the next step.*

S5 *If $\lambda'_u > \lambda'_\ell$, then let $\Delta = \Delta/2$ and $n_1 = n_1 + \Delta$, else (i.e., $\lambda'_u < \lambda'_\ell$) let $\Delta = \Delta/2$ and $n_1 = n_1 - \Delta$. Go to* **S3**.

Step S1 in the algorithm is optional because it is merely used to determine if dockers exist in $[n_0, n]$. While it provides a one-step solution to the constrained optimization problem when there is no docker, it only contributes to computational overhead when dockers exist. In the case where there is no docker, the binary search via docked trial optimizations can eventually yield the same conclusion as this simple trial optimization, albeit in more steps.

However, it has been shown[2] that the trial solution provides more information than mere indication of docker presence/absence: it also provides a bound on the location of the first docker. Properties of the docked trial solutions as given in the last theorem can also be further exploited to bound the docker locations. Below we make use of these facts to improve on the algorithm efficiency, after an analysis of the complexity of the above algorithm.

4.2 Complexity of the basic forward progression algorithm

The above algorithm presumes known D-R relations for the video units to be encoded. Thus to use this algorithm, we first need to generate these D-R relations, which typically amounts to quantization of each macroblock in each video unit with all possible quantizer scales. Assume that, in addition to generating the D-R relations, we further compute and sort the "singular Lagrange multiplier values"[11] for these video units. (A singular Lagrange multiplier for video unit m is a number λ for which there is more than one solution to the problem $min_{Q(m)}\{D(m) + \lambda b(m)\}$, i.e., there is 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.) Under this assumption, we analyze the complexity of the above algorithm.

Steps S1, S3, and S4 are where the complexity lies. Assume there are Λ singular Lagrange multipliers per video unit. Λ is upper-bounded by the product of the number of macroblocks in each video unit and the number of selectable quantizers for each macroblock. A pass over S1, S3, or S4 requires, at worst, search among $(\Delta + 1)\Lambda$ singular Lagrange multipliers (with different numerical values for Δ in S1 and S3/S4), at a complexity on the order of $log_2[(\Delta + 1)\Lambda]$ steps employing binary search. For S1, each search step requires up to the order of $\Delta + 1$ additions to sum up the total rate and 2 comparisons with the constraint. For the docked trial optimization in S3 or S4, each search step may also require up to the order of $\Delta + 1$ additions to compute the rates, plus up to the order of $\Delta + 1$ comparisons with the rate constraints. The checking of Properties (i), (ii) and (iii) requires up to the order of $n - n_1$ additions to compute the rates and $2(n - n_1)$ comparisons with the rate constraints.

Let $K = \log_2(N + 1)$. Then for the solution of d_1 and λ'_{d_1}, the worst-case complexity, denoted $W_s(N + 1)$, is on the order of

$$\begin{aligned} W_s(N+1) &= (N+3)\log_2[(N+1)\Lambda] + \sum_{i=K-1}^{0} 4(2^i+1)\log_2(2^i\Lambda) + \sum_{i=K-1}^{0} 3(N+1-2^i) \\ &= \mathcal{O}\left(8(N+1)\log_2(N+1) + 5(N+1)\log_2\Lambda\right) \end{aligned} \tag{11}$$

arithmetic operations, which is an order of magnitude lower than the worst-case complexity of an earlier algorithm.[2] Analysis of the overall algorithm can be carried out using similar techniques.

4.3 Acceleration of the forward progression algorithm

Assume there is at least one docker in $[n_0, n]$, where n_0 is as defined in the earlier algorithm statement. A notion that is of use to improving the algorithm efficiency is that of the *anchor point*[2] in the trial solution over $[n_0, n]$, defined as the last video unit that commits overflow that precedes the first underflow, or the last video unit that commits underflow that precedes the first overflow, whichever condition holds. If only overflows or only underflows occur over $[n_0, n]$, then it is the last video unit where such occurs. The idea of anchor points is illustrated in Fig. 3. We expand the idea and define the *extended anchor point* as the first-comer of the following two: (i) the anchor point, and (ii) the last video unit that touches the rhs (resp. lhs) rate boundary that precedes the first underflow (resp. overflow), or the last video unit that commits overflow (resp. underflow) that precedes the first touching of the lhs (resp. rhs) rate boundary, whichever condition holds. We have the following result (cf. Lemma 2 in Chen and Lin[2]).

LEMMA 4.2. **(The Anchor Position Property)** *The extended anchor point in the trial solution is located at or after the first docker point in* $[n_0, n]$.

From the above property, the binary search for the first docker need be conducted only over the *extended anchor subsequence* $[n_0, a]$, where a denotes the extended anchor point, instead of over the whole $[n_0, n]$. In addition, from the result concerning Condition (iii)(b) in Theorem 4.1, we can show that, for an upper (resp. lower) docked trial solution, if the rate boundary that is violated at the anchor point v is the lhs (resp. rhs) boundary, then there exist at least one upper and one lower docker in the *anchor subsequence* $[n_0, v]$. We thus obtain a modified algorithm as follows.

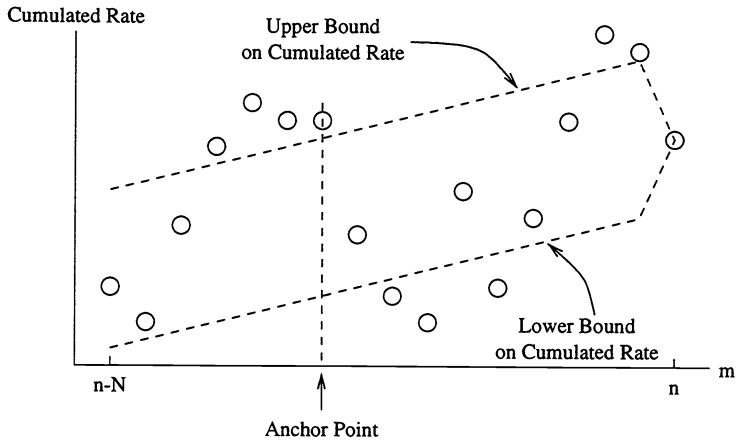

Figure 3: Concept of anchor points.

ALGORITHM 2. **(The Anchor-Assisted Forward Progression Algorithm)**

S0 (Initialization) *Let* $n_0 = n - N$, $\Delta = N$, *and* $i = 1$.

S1 (Trial optimization, optional) *Do trial optimization over* $[n_0, n_0 + \Delta]$. *If all the rate constraints associated with video units* $[n_0, n_0 + \Delta]$ *are satisfied (i.e., if (1) holds* $\forall k \in [n_0, n_0 + \Delta]$*), then exit. Otherwise, locate the extended anchor point* a *and let* $\Delta = a - n_0$.

S2 *Let* $\Delta = \Delta/2$ *and* $n_1 = n_0 + \Delta$.

S3 (Upper docked trial optimization) *Do upper docked trial optimization over* $[n_0, n_1]$. *Check if there exist* $p, q \in [n_0, n)$ *that satisfy Properties (i), (ii), and (iii) in the First-Docker Identification Theorem. If so, then we have* $d_i = p$ *and* $\lambda'_{d_i} = \lambda'_u$; *locate the anchor point* v *in the upper docked trial solution, let* $n_0 = d_i + 1$, $\Delta = v - n_0$, *and* $i = i + 1$, *and go to* **S1**. *If not, then go to the next step.*

S4 (Lower docked trial optimization) *Do lower docked trial optimization over* $[n_0, n_1]$. *Check if there exist* $p, q \in [n_0, n)$ *that satisfy Properties (i), (ii), and (iii) in the First-Docker Identification Theorem. If so, then we have* $d_i = p$ *and* $\lambda'_{d_i} = \lambda'_\ell$; *locate the anchor point* v *in the lower docked trial solution, let* $n_0 = d_i + 1$, $\Delta = v - n_0$, *and* $i = i + 1$, *and go to* **S1**. *If not, then go to the next step.*

S5 *If* $\lambda'_u > \lambda'_\ell$, *then let* $\Delta = \Delta/2$ *and* $n_1 = n_1 + \Delta$ *and go to* **S3**, *else (i.e.,* $\lambda'_u < \lambda'_\ell$*) do the following. Locate the extended anchor point* a_u *in the upper docked trial solution and the extended anchor point* a_ℓ *in the lower docked trial solution. Let* $\Delta = \min(a_u - n_0, a_\ell - n_0)/2$ *and* $n_1 = n_0 + \Delta$. *Go to* **S3**.

Further acceleration of the algorithm is possible. For example, we may selectively employ the present algorithms or an earlier algorithm (which is based on repeated trial optimizations over successive anchor subsequences),[2] depending on the value of Δ in S1. We may also be able to arrange the sequence in which S3 and S4 are carried out, dynamically according to the properties of some intermediate algorithm results, so as to maximize the number of occasions that only one of them is required. We may further be able to bound the docker locations more elegantly than the ways described in the above algorithms.

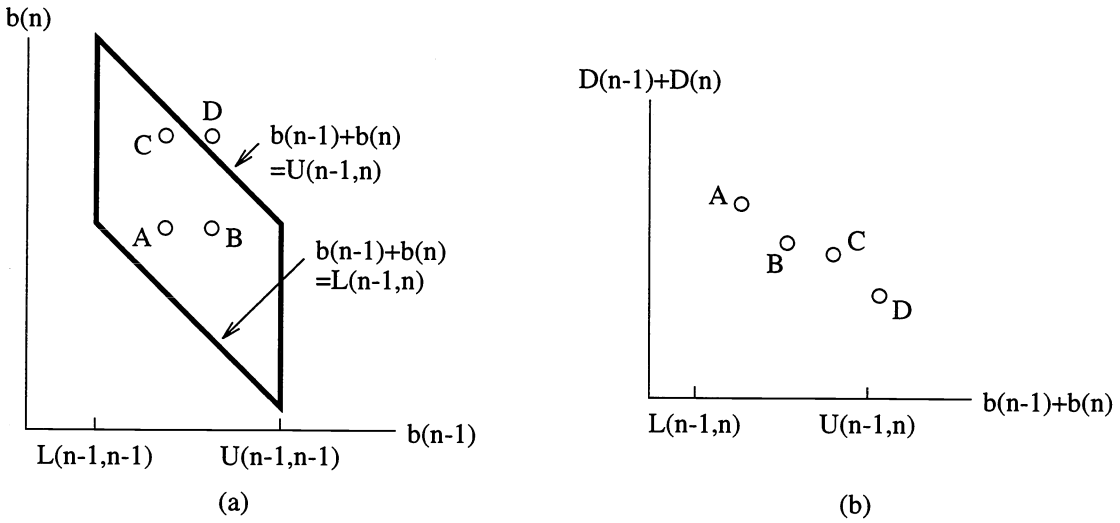

Figure 4: Example illustrating a mechanism underlying the segmental uni-slope solution's potential suboptimality in discrete rate allocation. (a) A typical region of permitted rate combinations when coding delay = 1 video unit and four out of all selectable rate combinations denoted A, B, C, and D. (b) The aggregate D-R relation of the four rate combinations.

4.4 Considerations in discrete rate allocation

In discrete rate allocation, we may not be able to impose the exact equalities in (6), (8), (9), and (10), but have to allow for deviation due to rate granularity. This is relatively easy by our use of the singular Lagrange multipliers, which can find the solution that has minimum deviation from the rate boundary.[11]

Somewhat more complicated to handle is the fact that the Lagrange-multiplier method only finds solutions on the convex hull of the D-R relations. To examine the situation more closely, recall that our solution observes a "segmental uni-slope property," that is, the prime Lagrange multipliers in the solution are equal over each docker subsequence $(d_i, d_{i+1}]$ (where $i = 0, 1, \cdots, A$) or equivalently, the (original) Lagrange multipliers in the solution are all zero except $\lambda_{d_i - n + N}$ (where $i = 1, \cdots, A$). For ease of reference, term our solution as the *segmental uni-slope solution* and the associated optimization approach the *segmental uni-slope optimization.*

Consider the simple case where there is no docker in $[n - N, n)$. Cases where dockers exist exhibit corresponding characteristics. With no dockers in $[n - N, n)$, the prime Lagrange multipliers in our solution are all equal, or equivalently, the (original) Lagrange multipliers in the solution are all zero except λ_N. Therefore, the solution obtained by the algorithms is located on the convex hull of the *aggregate* D-R relation of the video units $[n-N, n]$, i.e., on the convex hull formed by all possible pairs of $\sum_{k=n-N}^{n} D(k)$ and $\sum_{k=n-N}^{n} b(k)$. However, what the constrained optimization (2) really looks for is the optimum over all possible combinations of $b(k)$, $k = n - N, \cdots, n$, i.e., the optimum over the N-dimensional function of $\sum_{k=n-N}^{n} D(k)$ vs. $\{b(k), k = n - N, \cdots, n\}$. The two may not coincide in the case of discrete rate allocation (although in continuous rate allocation, they do, as has been shown in the foregoing discussion).

For example, when $N = 1$, we may have the situation illustrated in Fig. 4. The circles in the figure represent several selectable rate combinations for video units $n - 1$ and n. Fig. 4(b) shows that Points B and D are on the convex hull of the aggregate D-R relation while Point C is not. Indeed, Point B gives the segmental uni-slope solution. However, the truly optimal solution is at Point C. Note that this situation may happen even if both the component D-R relations, i.e., $D(n-1)$-$R(n-1)$ and $D(n)$-$R(n)$, are convex. In fact, when both the component

D-R relations are convex, the three-dimensional plot of $\sum_{k=n-1}^{n} D(k)$ vs. $\{b(k), k = n-1, n\}$ is convex and the truly optimal solution is characterized by two prime Lagrange multipliers of unequal values. Regrettably, the segmental uni-slope optimization does not find this solution unless $n-1$ is a docker (and hence a change in prime Lagrange multiplier value there is possible). But a Lagrange-multiplier solution for the truly optimal solution in the above situation appears quite complicated. Thus we shall be content with the segmental uni-slope solution for the time being. It should be of interest to obtain quantitative characterization of the potential suboptimality in the segmental uni-slope solution, which is left for potential future work.

When an improved solution than the segmental uni-slope solution is desirable, a tree/trellis search over the D-R relations in the proximity of the latter solution can be conducted.[12–15,2] This is applicable to the situation where the D-R relations of the video units are convex as well as the situation where these D-R relations are non-convex to start with.

There is a condition when the segmental uni-slope solution is guaranteed optimal in discrete rate allocation. That is when the D-R relations of the video units are all convex and the selectable rates for the video units are all uniformly spaced with the same granularity. We omit the proof here.

5 CONCLUSION

We studied optimal bit allocation for delayed-coding of a sequence of video units under multiple rate constraints which may arise from finite codec delay, finite channel capacity, and finite codec buffer sizes. An approach employing multiple Lagrange multipliers was adopted and two efficient solution algorithms were derived. The algorithms are based on efficient ways of searching for a particular kind of points in time called the docker points. By identification of the docker points the multiple Lagrange-multipliers optimization problem is decomposed into a series of one Lagrange-multiplier problems readily solvable by known techniques.

The solution is optimal when the distortion-rate relations of the video units are convex and the selectable rates of the video units are uniformly spaced with the same granularity. When these conditions do not hold, the Lagrange-multiplier solution may be suboptimal, but can be improved or optimized by a search about the solution. Simulations results will be presented in the full paper.[19]

The problem has been studied under the assumption of independent coding, that is, the D-R relations of later video units in a delayed-coding window do not depend on how earlier video units in the window are encoded. However, the algorithms can be employed in dependent coding to obtain suboptimal solutions. In addition, the results are applicable to both constant-rate and variable-rate transmission environments.

6 ACKNOWLEDGMENT

This work was supported in part by National Science Council of ROC under grant NSC 86-2221-E-009-019.

7 REFERENCES

[1] D. W. Lin, M.-H. Wang, and J.-J. Chen, "Optimal delayed-coding of video sequences subject to a buffer-size constraint," in *SPIE vol. 2094, Visual Commun. Image Process. '93*, pt. 1, pp. 223–234, Nov. 1993.

[2] J.-J. Chen and D. W. Lin, "Optimal bit allocation for coding video signal over ATM networks," revised version submitted to *IEEE J. Select. Areas Commun.*, Sep. 1996.

[3] J.-J. Chen and D. W. Lin, "Optimal bit allocation for video coding under multiple constraints," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 403–406, Sep. 1996.

[4] 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.

[5] E. P. Rathgeb, "Policing of realistic VBR video traffic in an ATM network," *Int. J. Digital and Analog Commun. Systems,* vol. 6, pp. 213–226, 1993.

[6] "Video codec for audiovisual services at $P \times 64$ kbit/s," ITU-T Recommendation H.261.

[7] "Information technology — generic coding of moving pictures and associated audio: video," ISO/IEC 13818-2 and ITU-T Recommendation H.262.

[8] "Video coding for low bitrate communication," draft ITU-T Recommendation H.263, Oct. 1995.

[9] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard.* New York: Van Nostrand Reinhold, 1993.

[10] "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.

[11] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.

[12] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximation," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 26–40, Jan. 1994.

[13] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multi-resolution and MPEG video coders," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 533–545, Sep. 1994.

[14] C.-Y. Hsu and A. Ortega, "Joint channel encoder and VBR channel optimization with buffer and leaky bucket constraints," in *Symp. Multimedia Commun. and Video Coding*, Brooklyn, NY, Oct. 1995.

[15] C.-Y. Hsu, A. Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," revised version submitted to *IEEE J. Select. Areas Commun.*, Oct. 1996.

[16] A. Ortega, "Optimal bit allocation under multiple rate constraints," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 1996.

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

[18] D. G. Luenberger, *Optimization by Vector Space Methods.* New York: Wiley, 1969.

[19] D. W. Lin and J.-J. Chen, "Efficient bit allocation under multiple constraints on cumulated rates for delayed video coding," full-length paper in preparation.