Rate-Distortion Optimized Frame Type Selection for MPEG Encoding

Jungwoo Lee, Member, IEEE, and Bradley W. Dickinson, Fellow, IEEE

Abstract—In this paper, we present an algorithm for joint optimization of anchor frame separation and bit allocation for motion-compensated video coders. The anchor frame separation is optimized in the sense that the distortion is minimized under a bit budget constraint. At the same time, the quantization for each frame in a group of pictures is also optimized in an operational rate distortion sense. The optimal anchor frame separation does depend on the quantization of each frame so that the two optimization problems cannot be separated. A Lagrange multiplier approach can be used to obtain the optimal solution if we assume that the rate-distortion curve is convex. Heuristic algorithms based on simulated annealing and greedy trellis selection are also presented to reduce the computational complexity.

Index Terms—Image coding, joint optimization, MPEG, rate control, simulated annealing

I. INTRODUCTION

T
HE authors recently proposed temporally adaptive motion interpolation (TAMI) [1]. The basic idea of the previous work is that the MPEG frame type (or the anchor frame separation) is dynamically adapted according to the motion activity of the input video. An anchor frame is the frame from which the current frame is predicted in MPEG motion compensation. The algorithm proposed in this paper is an optimized version of the previous work. It also incorporates an optimal bit allocation method in [2] for MPEG coders [3]. The optimal bit allocation is based on a Lagrange multiplier method that minimizes the distortion under the fixed bit budget constraint.

In MPEG encoding, there are three different frame types (I, P, and B frames). An I frame is an intra frame, which means that it is coded by itself without any motion compensation. A P frame is a predictive frame which means that it is coded with motion compensation from a previous anchor frame. A B frame is an interlative frame which is coded with motion compensation both from a previous anchor frame and from a future anchor frame. In conventional MPEG encoding, the arrangement of the three frame types (called the group of pictures or GOP) is fixed, and it is typically of the form IBPBBBP... The idea of this paper is that the arrangement is dynamically adapted in order to optimize compression by individually selecting the frame type for each frame. By design, the frame type adaptation is within the scope of MPEG-1 and MPEG-2 standards. One of the goals in this paper is to present an algorithm that optimizes the frame type selection in a rate-distortion sense. This optimal scheme will be especially useful for nonreal time MPEG applications such as CD-ROM and digital video disk (DVD) systems because it offers the best frame type selection and the best bit allocation.

Rate-distortion theory [4] deals with minimization of distortion under a fixed channel bit rate constraint. The problem can be interpreted as one of minimizing channel bit rate subject to a given distortion. In this paper, we are trying to solve the minimization problem when there are many signal blocks. The key issue then becomes how to efficiently allocate bits among different signal blocks which are image frames in this case. This has been studied extensively especially in the vector quantization (VQ) area [5]–[7].

In MPEG-1 or MPEG-2 encoders, the number of quantizers (quantization step sizes) for discrete cosine transform (DCT) coefficients is typically fixed. Efficient bit allocation among signal blocks when the number of quantizers is finite has been studied in [8], [5], and [6]. In this paper, we also assume that the number of quantizers is fixed as is required by the MPEG standard. Our goal is to jointly find optimal frame types and optimal quantizer choices for each frame in a GOP.

The constrained optimization problem in this paper is solved by a Lagrangian multiplier technique. This problem is an optimal bit allocation problem where the constraint is the fixed total bit budget. The basic idea of the Lagrange multiplier technique is that the constrained minimization problem is converted to an unconstrained minimization problem which is easier to solve. It can be easily shown that the solution of the new unconstrained problem is also the solution for the original constrained problem if the bit rate solution in the unconstrained problem happens to be the same as the bit budget in the original constrained problem [6]. The proof of this relationship for the joint optimization problem of GOP structure and bit allocation in this paper will be given in the next section.

Another important concept used in this paper is the trellis-based quantizer optimization technique that was proposed in [2]. The trellis is used because the Lagrangian cost of one frame depends on the quantizer of its previous anchor frame. The selection of the optimal quantizer set corresponds to the problem of determining the optimal path in a quantizer trellis diagram. One example of such a diagram is shown in Fig. 1. Fast algorithms such as Viterbi algorithms...
Fig. 1. An example of GOP structure with corresponding quantization trellis diagram for the case of three quantization values. Note that the quantizer for the first I frame is fixed because it was already determined in the previous GOP bit allocation.

exist to find the optimal path in a trellis. But it cannot be used in the MPEG encoding case because one frame can be dependent on multiple previous anchor frames. If, for example, the GOP structure is IPP, the second P frame is dependent not only on the first P frame but also on the first I frame. If the dependency depth is longer than one stage, the Viterbi algorithm cannot be used because the algorithm assumes that each stage of the trellis is independent of others.

To avoid an exhaustive trellis search, a heuristic scheme based on the monotonicity was introduced [2]. It was empirically observed that, for most video material, the Lagrangian cost of one frame increases as the quantizer of its previous anchor frame becomes coarser than the quantizer that generates minimum Lagrangian cost of the previous frame. Using the monotonicity assumption, some of the trellis paths can be discarded in the optimal path search. A more detailed description of the monotonicity will be given later in this paper. The trellis search algorithm in this paper is based on the algorithm proposed in [2]. The difference between them is that the algorithm of this paper deals with dynamic GOP structure whereas the algorithm in [2] assumes a fixed GOP structure. The algorithm in [2] is applied to a set of frames that are between two neighboring anchor frames (e.g., IBBP). The path pruning rule based on the monotonicity is applied twice; one for I-P pair and the other for I-B-B-P combination. The key steps of the algorithm are similar to those of the generalized algorithm for dynamic GOP structure which will be described later.

This paper is structured in the following way. In Section II, we define the problem of minimizing the distortion with respect to the anchor frame position and the quantizer. The conversion from the constrained minimization problem to the unconstrained one by using Lagrange multiplier is also discussed. Section III deals with determining an optimal quantizer trellis for given Lagrange multiplier and anchor frame positions. It also discusses a fast optimal algorithm based on the monotonicity of the Lagrangian cost with respect to the quantizer. In Section IV, we present an algorithm that optimizes the number of anchor frames and their positions. In the development of the algorithm, we make the assumption that the Lagrangian cost has only one minimum with respect to the number of anchor frames. In Section V, the complete algorithm for frame type and quantization optimization is summarized. It also briefly describes a method to obtain the desired Lagrange multiplier. Heuristic methods to reduce the complexity of the quantizer trellis search are discussed. Section VI discusses the simulated annealing approach for optimizing anchor frame positions. A brief discussion of the stopping criterion and the annealing schedule is also given. In Section VII, the complexities of the exhaustive search and the heuristic algorithms are analyzed. The complexity of the fast optimal algorithm is also determined. Experimental results are presented in Section VIII. Section IX presents a summary and concludes the paper.

II. PROBLEM FORMULATION

In MPEG [3], the typical range of the DCT coefficient quantization parameter \( q \) is \( 1 \leq q \leq 31 \). In this formulation, we limit the number of allowable quantization parameters. Let us denote the set of \( L_q \) values of the allowable quantization parameter by \( S(L_q) \). Note that \( S(L_q) \) is a subset of \( \{1, 2, \ldots, 31\} \). For example, when \( L_q = 3 \), \( q \) may take values in \( S(L_q) = \{1, 15, 31\} \). Let us also denote the quantization parameter for frame \( i \) by \( q_i \).

The constrained optimization problem to be solved here is to determine jointly the number and the position of reference (anchor) frames in a GOP, and to choose the quantization parameter \( q_i \) for each frame in the GOP such that the total distortion is minimized subject to a maximum total bit budget constraint. Let us define the following terms.

- \( N \) Number of P frames used in a GOP.
- \( N_a \) \{0, 1, \ldots, M - 1\}.
- \( P \) Vector of positions of the \( N \) reference frames.
- \( P_a(N) \) Set of all possible vectors of positions of \( N \) reference frames \( \{p_0, p_1, \ldots, p_{N-1}\} | 1 \leq p_0 \leq p_1 \leq \cdots \leq p_{N-1} \leq M - 1 \).
Vector of quantization parameters for each frame in a GOP. Note that $Q = (q_0, q_1, \ldots, q_{M-1})$ where $q_i \in \mathcal{S}(L_q)$ ($i = 0, 1, \ldots, M - 1$).

Set of all possible vectors of quantization parameters of $M$ frames:

$\{ (q_0, q_1, \ldots, q_{M-1}) | q_i \in \mathcal{S}(L_q) \}$.

$D(N, P, Q)$ Distortion with respect to $N$, $P$, and $Q$.

Rate (in bits) with respect to $N$, $P$, and $Q$.

The problem to solve is that of finding the optimal number ($N$) and positions ($P$) of reference frames with their associated optimal quantizers $Q$. Stated mathematically, this is formulated as the following minimization problem:

$$\min_N \min_P \min_Q D(N, P, Q) \text{ such that } R(N, P, Q) \leq R_{\text{budget}}.$$  

(1)

Instead of solving (1), let us consider a related unconstrained optimization problem. This is formulated by the generalized Lagrange multiplier method for resource allocation problems, as introduced by Everett [9]. Let us introduce the Lagrangian cost function corresponding to the Lagrange multiplier $\lambda \geq 0$, the motion interpolation structure ($N$, $P$), and the quantizer $Q$:

$$J(N, P, Q, \lambda) = D(N, P, Q) + \lambda R(N, P, Q).$$  

(2)

Then (1) can be replaced by a parametrized family of unconstrained minimization problems

$$\min_N \min_P \min_Q [D(N, P, Q) + \lambda R(N, P, Q)].$$  

(3)

The relationship between (1) and (3) is that if we obtain the solution $(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$ for (3) for some $\lambda \geq 0$, and $R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$ happens to be $R_{\text{budget}}$ of the constrained problem (1), the solutions for the problems are identical. This can be easily proved by extending the result in [6].

**Theorem 1:** If $(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$ is the solution of the unconstrained problem (3) corresponding to some fixed value of $\lambda$, then it is also the solution to the constrained problem (1) for the particular case of $R_{\text{budget}} = R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$; i.e., for this budget $R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) = R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$, $N_{\lambda}^* = N_{\lambda}^*$, and $P_{\lambda}^* = P_{\lambda}^*$ where $N_{\lambda}^*$, $P_{\lambda}^*$, and $Q_{\lambda}^*$ comprise a solution to (1).

**Proof:**

$$J(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*, \lambda) \leq J(N, P, Q, \lambda)$$  

(4)

$$D(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) + \lambda R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) \leq D(N, P, Q) + \lambda R(N, P, Q).$$  

(5)

$$D(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) - D(N, P, Q) \leq \lambda (R(N, P, Q) - R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*))$$  

(6)

$$D(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) - D(N, P, Q) \leq \lambda (R(N, P, Q) - R_{\text{budget}}).$$  

(7)

Since (7) holds for all $N \in N_{\lambda}^*, P \in P_{\lambda}^*$, and $Q \in Q_{\lambda}^*$, it also holds for the subsets $N \subset N_{\lambda}^*, P \subset P_{\lambda}^*, Q \subset Q_{\lambda}^*$ which satisfy $R(N, P, Q) \leq R_{\text{budget}}$. That is

$$R(N, P, Q) \leq R_{\text{budget}} \text{ for } N \in N_{\lambda}^*, P \in P_{\lambda}^*, Q \in Q_{\lambda}^*.$$  

(8)

Thus from (7) and (8), $\lambda \geq 0$, we have

$$D(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) - D(N, P, Q) \leq 0$$  

(9)

which is satisfied for all $N \in N_{\lambda}^*, P \in P_{\lambda}^*, Q \in Q_{\lambda}^*$. That is, $(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$ also satisfies the original constrained optimization problem (1) for the given budget constraint, and $R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*) = R(N_{\lambda}^*, P_{\lambda}^*, Q_{\lambda}^*)$, $N_{\lambda}^* = N_{\lambda}^*$, and $P_{\lambda}^* = P_{\lambda}^*$.

**III. DETERMINING OPTIMAL QUANTIZERS**

In this section, we consider solving for best quantizers, assuming that $N$ and $P$ are fixed. The basic idea in this section is based on a recent result for optimal bit allocation in dependent coding environments in [2]. In the paper, a trellis search technique was introduced to obtain an optimal quantization solution for an MPEG-like predictive coding scheme, where the coding of one frame depends on the coding results of its previous frames. For the correct value of $\lambda$ that meets the bit budget constraint, the problem we need to solve here is

$$\min_{Q_1 \in \mathcal{Q}_1, \ldots, Q_{M-1} \in \mathcal{Q}_{M-1}} J(q_1, q_2, \ldots, q_{M-1}) = D(q_1, q_2, \ldots, q_{M-1}) + \lambda R(q_1, q_2, \ldots, q_{M-1})$$

(10)

where $J(q_1, q_2, \ldots, q_{M-1}) = D(q_1, q_2, \ldots, q_{M-1}) + \lambda R(q_1, q_2, \ldots, q_{M-1})$ is the total Lagrangian cost as a function of quantizers for frame 1 through $M - 1$.

If each frame is independently coded as an I frame (e.g., JPEG video), then the total cost can be decomposed as

$$J(q_1, q_2, \ldots, q_{M-1}) = J_1(q_1) + J_2(q_2) + \cdots + J_{M-1}(q_{M-1}).$$

where $J_i(q_i) = D_i(q_i) + \lambda R_i(q_i)$ is the Lagrangian cost of frame $i$. In this case, the well-known result of optimal bit allocation for independent coding can be applied [6], where the basic paradigm is that each frame can be independently optimized at a rate-distortion (R-D) point with a constant slope which is the same as the Lagrange multiplier $\lambda$.

When the coding of one frame is dependent on previous frames (e.g., MPEG or DPCM), the total cost $J$ in (10) cannot be decomposed as in (11). Let us introduce a dependent Lagrangian cost for frame $i$:

$$J_i(q_1, q_2, \ldots, q_i) = D_i(q_1, q_2, \ldots, q_i) + \lambda R_i(q_1, q_2, \ldots, q_i)$$

(12)

where $i = 1, 2, \ldots, M - 1$. The problem (10) becomes the following unconstrained problem

$$\min_{Q_1 \in \mathcal{Q}_1, \ldots, Q_{M-1} \in \mathcal{Q}_{M-1}} [J_1(q_1) + J_2(q_1, q_2) + \cdots + J_{M-1}(q_1, q_2, \ldots, q_{M-1})].$$

(13)

In this case, the quality factor $\lambda$ is not the same as the slope of the optimal R-D operating point at each frame, as can be seen in Fig. 2(a) and (b). This is because the set of available R-D operating points for one frame [see Fig. 2(b)] may depend on the choices of quantization parameter of other frames [see Fig. 2(a)].

The above result implies that to obtain the optimal solution for (13), we need to find R-D points $(R_i(q_1, q_2, \ldots, q_i), D_i(q_1, q_2, \ldots, q_i))$ for all possible trellis paths as in Fig. 1, which becomes exponentially complex as the dependency tree
Fig. 2. Operational R-D characteristics of two frames in dependent coding framework. (a) R-D curve of frame 1. (b) R-D curve of frame 2 (dependent frame). The curve of frame 2 is dependent on the choice of the quantizer for frame 1.

Fig. 3. Pruning conditions exploiting monotonicity. (a) Any path which lies above the minimum path can survive. Those below it can be pruned out. (b) The same pruning rule as in (a) can be used for these paths having common source node.

depth \( i \) becomes larger. Notice that it is the data generation phase, not the trellis search phase, which is computationally complex. To reduce this computational burden, we employ a method which will avoid the need to generate unnecessary R-D data, while retaining optimality. This method exploits the fact that the quality of prediction for a dependent frame monotonically depends on the fineness of the quantizer for its parent frame from which the prediction is originated. Let us denote \( i \) and \( j \) quantizer parameters for the independent and dependent frames. The monotonicity holds if

\[
J_2(i, j) \leq J_2(i', j) \quad \text{for} \quad i \leq i'.
\]  

(14)

In other words, the prediction residual of a future dependent frame always tends to increase if the parent frame on which the prediction is based is encoded with coarser quantization. It was reported in [2] that the monotonicity assumption is true for most ordinary video material.

Based on this monotonicity, two kinds of pruning in the cost generation stage are used. The first pruning rule deals with the pruning of the branches that merge into a common destination node [see Fig. 3(a)], which can be easily proved by the monotonicity assumption [2]. If \( J_1(i) + J_2(i, j) < J_1(i') + J_2(i', j) \) for any \( i < i' \), then the \( (i', j) \) branch cannot be part of the optimal path and can be pruned out. The second pruning rule is associated with pruning of branches that originate from a common source node [see Fig. 3(b)]. If \( J_2(i, j) < J_2(i, j') \) for any \( j < j' \), then the \( (i, j') \) branch cannot be part of the optimal path and can be pruned out. This can also be easily proved by the monotonicity assumption.

The algorithm that determines the quantizer trellis for arbitrary GOP structure is described as follows, which is generalized from the trellis determination algorithm in [2]. It is assumed in [2] that the GOP structure is fixed. On the other hand, the number of B frames between two anchor frames (I or P frame) can be variable in this paper. Note that the quantization of the first I frame is always fixed, because it is determined in the previous GOP optimization (see Fig. 1).
Algorithm 1:
1) Compute the Lagrangian cost $J$ of the next anchor frame for all possible quantizer branches.
2) (Monotonicity) Prune suboptimal I-P or P-P paths using the two pruning rules.
3) For every surviving path, find the quantizer sets of intervening B frames that minimize $\sum J(B_k)$.
4) (Monotonicity) Eliminate suboptimal I-B-B-P or P-B-B-B-P using the two pruning rules.
5) For all remaining paths, repeat Step 1 to 4 for next (P-B-B-P) or (P-B-B-I) sets.

The reduction of computation is dependent on the Lagrange multiplier $\lambda$; a smaller $\lambda$ leads to less computation. For example, if $\lambda$ is small, the algorithm focuses on reducing the distortion $D$ rather than on the bits $R$ as can be immediately seen in the Lagrangian cost definition, $D+\lambda R$. It follows that a finer quantizer tends to be chosen, and that the minimum paths in Fig. 3(a) and (b) tend to be located in upper nodes. This means that more paths can be pruned and that less computation is needed.

IV. REFERENCE FRAME POSITIONING

To solve for the optimal number and positions ($N$ and $P$) of the reference frames in (3), we assume unimodality of the Lagrange cost with respect to $N$, which was true for all the video sequences we tested. In other words, Lagrangian cost as a function of $N$ tends to have only one minimum as shown in Fig. 4 for most video material (unimodality). Algorithm 1:

To find the best reference frame positions $P$ given $N$, exhaustive search is used since it appears that in general there is no simple relationship between $P$ and the corresponding Lagrange cost. Later in this paper, we develop a heuristic algorithm based on simulated annealing to reduce the computational complexity of the search for (near-)optimal reference frame positions, $P$.

V. COMPLETE ALGORITHM

In this section, we try to combine the individual algorithms of the previous sections. The combined algorithms consist of two parts. For a given fixed $\lambda$, the first part deals with minimizations of $N$, $P$, and $Q$. The second part is an algorithm to iteratively determine optimal $\lambda^*$ that meets the budget constraint $R_{\text{budget}}$. Before starting the main algorithm, motion vectors between every possible pair of reference frames are generated using telescopc search which was well described in [1].

In telescopc search, the motion vector for a macroblock of a previous frame is used as center position (prediction) around which the motion search for the same macroblock position of a current frame is performed. One advantage of the search technique is that the effective search range can be increased without increasing the actual search range. Let us denote the distance between the current frame and its previous reference frame by $T$, the effective search range by $E$, and the actual search range by $A$. Then we have $E = TA$. In other words, large effective search range can be obtained by cascading the telescopc search over multiple frames. This initialization step reduces the computational complexity of the proposed algorithm. The number of sets of motion vectors to be computed is $2^M \choose 2$, which is 210 when the GOP size $M$ is 15. The motion vectors are computed at the beginning of the algorithm in order to take advantage of the telescopc search which can reduce the motion vector computation significantly [1].

The first part of the algorithm (Algorithm 2) searches for optimal reference frame number and position and quantizer sets for given $\lambda$. The Lagrange multiplier $\lambda$ is updated by the second part of the algorithm, Algorithm 3.

Algorithm 2:
1) (Initialization) Previous Lagrangian cost $J_p \leftarrow 0$; previous cost increment $\delta_p \leftarrow 0$; $N \leftarrow 0$.
2) Run Algorithm 1 to compute minimum $J(N,P)$ and optimal quantization trellis path $T(N,P)$ for all possible $P$.
3) Find $P_{\text{min}}$ that minimizes $J(N,P)$, i.e., $J(N,P_{\text{min}}) \leq J(N,P)$. Compute Lagrangian cost increment $\delta \leftarrow (J(N,P_{\text{min}}) - J_p)$.
4) If $(\delta \geq 0$ and $\delta_p < 0)$ then $N^* \leftarrow N_p$; $P^* \leftarrow P_p$.
5) Stop.
   else $J_p \leftarrow J(N,P_{\text{min}})$; $N_p \leftarrow N$; $P_p \leftarrow P(N,P_{\text{min}})$.$N \leftarrow (N + 1)$
   Go to Step 2.

A. Bisection Algorithm for Lagrange Multiplier

Let us consider bit rate as a function of $\lambda$, i.e., $R = R(\lambda)$. It can be shown [6] that the rate as a function of $\lambda$, $R(\lambda)$, is monotonically decreasing. Hence, there exists a fast binary
search method for \( \lambda \) by exploiting the monotonicity [6]. This algorithm has to be started with two initial values of \( \lambda \). The number of iterations of the algorithm may depend on these values. We assume that we can choose \( \lambda_L \) and \( \lambda_U \) such that

\[
R(\lambda_L) \leq R_{\text{unlq} \text{igt}} \leq R(\lambda_U).
\] (15)

If no such \( \lambda_L \) and \( \lambda_U \) exist, the problem is unsolvable because \( R_{\text{unlq} \text{igt}} \) is outside the range of achievable bit rate by the given set of quantizers. For fixed \( \lambda \), the solution we obtain for (3) is also the solution for (1) with the budget constraint \( R_{\text{unlq} \text{igt}} = R(\lambda) \). Therefore, to find the best solution for the unconstrained problem, we may iteratively change \( \lambda \) to find the maximum achievable \( R(\lambda) \) such that \( R(\lambda) \leq R_{\text{unlq} \text{igt}} \). It may be possible that \( R(\lambda) \) cannot be exactly matched to \( R_{\text{unlq} \text{igt}} \) because we only consider a finite number of operating points so that \( R(\lambda) \) takes in a discrete set of values. Algorithm 3 is a fast search method for \( \lambda \) using bisection algorithm. It is described as follows; more details are in [6] and [9]–[11].

**Algorithm 3:**

1. Pick \( \lambda_L \) and \( \lambda_U \) such that \( R(\lambda_L) \leq R_{\text{unlq} \text{igt}} \leq R(\lambda_U) \).
2. \( \lambda_{\text{next}} = \left| \frac{D_0(\lambda_L) - D_0(\lambda_U)}{R(\lambda_L) - R(\lambda_U)} \right| + c. \)
3. Run Algorithm 2 for \( \lambda = \lambda_{\text{next}} \).
4. If \( R(\lambda_{\text{next}}) = R(\lambda_L) \), then stop with \( \lambda^* = \lambda_U \).
5. If \( R(\lambda_{\text{next}}) > R_{\text{unlq} \text{igt}} \), \( \lambda_L \leftarrow \lambda_{\text{next}} \) else \( \lambda_U \leftarrow \lambda_{\text{next}} \).
6. Go to Step 2).

Here \( c \) is a small number (10^{-7} to 10^{-5}) that ensures that the algorithm converges to the lower bit rate solution when there is more than one solution. A typical number of iterations needed is about ten.

**B. Suboptimal Heuristic Algorithms for Quantizer Sequences**

Algorithm 1 dealt with finding an optimal quantizer sequence for given \( N \) and \( P \). To reduce computational complexity, it may not suffice to rely only on monotonicity in the trellis search. In this section, we will summarize a fast heuristic algorithm to find near-optimal quantizer sequences [2]. The following greedy heuristic is suggested: along with monotonicity, prune all branches except the lowest cost branch in the trellis. Since this heuristic allows us to select just one path, we can achieve suboptimal, but hopefully near-optimal, performance with much less computational complexity. Another advantage is that the reduction of computation is still possible even for large \( \lambda \), whereas the algorithm based on monotonicity works best when \( \lambda \) is low.

We described a heuristic for fast trellis path search. To reduce the computational further, this may be combined with simulated annealing to replace the \((N, P)\) search algorithm, which will be described in the next section.

**VI. SIMULATED ANNEALING FOR POSITION SEARCH**

In this section, we describe simulated annealing (SA) which will be used as a fast heuristic for obtaining \( P^* \) for given \( \lambda \) and \( N \). In Section IV we exploited unimodality of \( N \) to reduce the search space for \( N \), but exhaustive search was used to find optimal \( P \) because the behavior of the Lagrangian cost function with respect to \( P \) was almost unpredictable.

Let us consider the approximate number of iterations for each variable in the complete algorithm. In the minimization problem (3), experience has shown that rough estimates of the number of iterations for each variable in \((\lambda, N, P, Q)\) are ten for \( \lambda \), 5000 for \( N \) and \( P \), and 50 for \( Q \). The total number of iterations is the product of the three, 2.5 \times 10^6. Note that the computational complexity is dominated by the \((N, P)\) pair iteration. Hence, we are interested in fast suboptimal search methods for \( N \) and \( P \).

SA is guaranteed to approach a global minimum only after infinitely many iterations. Approximating the asymptotic behavior arbitrarily closely requires a number of transitions (site replacements) that is typically larger than the size of the solution space for most problems, leading to an exponential-time search algorithm [12]. Thus, SA is not suitable for solving for optimal answers. However, the asymptotic behavior of the algorithms can often be approximated in reasonable time. A polynomial-time heuristic returns a good suboptimal solution in most cases although the optimality has to be sacrificed.

**A. Site Selection and Replacement**

We are applying simulated annealing only to \( P \) search, because the unimodality of \( N \) can still be exploited to reduce the number of iterations to find correct \( N \), and experiments have also shown that the cost function has spiky behavior with respect to \( N \). Metropolis [13] developed an algorithm to select independent sample configurations according to the probability rule

\[
\frac{\text{prob}(x)}{\text{prob}(y)} = \frac{1}{Z} \exp(-\delta),
\]

for any specific value of \( T \), where \( Z \) is a normalization factor and \( T \) is a control parameter analogous to temperature.

In this algorithm, which is known as Metropolis algorithm, the acceptance probability from site \( i \) to \( j \) is \( e^{-\delta / T} \), where \( \delta = J_i - J_j \) is the Lagrangian cost difference between the two sites. We use the following site selection and replacement rules. **Site selection:** Pick position \( p_i \) to a new frame position \( p'_i \) which is randomly selected with uniform distribution in \( \{p_{i-1} + 1, p_{i-1} + 2, \cdots , p_{i+1} - 1\} \) (see Fig. 5). Here assume \( p_0 = 0 \) and \( p_N = M \).

The above replacement rule defines a neighborhood structure \( S(P) \), which is defined as

\[
S(P) = \{P' = (p'_0, p'_1, \cdots , p'_{i-1}) | \exists k \text{ such that } p_{i-1} < p'_i < p_{i+1} \text{ and } p'_j = p_j \text{ for all } j \neq i\}.
\] (16)
This choice of neighborhood structure is chosen because of an empirical observation that, for fixed \( N \), the Lagrangian cost changes smoothly when we change one reference position with all other positions fixed.

The simulated annealing algorithm is summarized as follows, using the following notations.

- **rep** and **maxrep**: Number of iterations for each \( T \) and its maximum value.
- **succ** and **maxsucc**: Number of successful acceptances and its maximum value.
- **temp** and **maxtemp**: Number of temperature iterations and its maximum value.

**Algorithm 4:**

1. (Initialization) Choose initial solution \( P \), initial temperature \( T_0 \), and temperature reduction function \( \alpha \).
2. (Site selection) Choose a site \( p_i \), \( 0 \leq i < N \) randomly with uniform probability.
3. (Site replacement) With uniform probability, choose new value \( p'_i \) randomly which satisfies \( p_{i-1} < p'_i < p_{i+1} \), and set \( p'_j = p_j \) for all \( j \neq i \).
4. Compute cost difference \( \delta = J(P') - J(P) \).
5. (Acceptance decision)
   - If \( \delta < 0 \), then \( P \leftarrow P' \) and \( \text{succ} \leftarrow \text{succ} + 1 \).
   - else generate random \( r \) uniformly in the range \((0, 1)\)
     - if \( r < \exp(-\sigma(T)/T) \), then \( P \leftarrow P' \) and \( \text{succ} \leftarrow \text{succ} + 1 \).
6. If \( \text{succ} \geq \text{maxsucc} \) then go to Step 8; else go to Step 7.
7. If \( \text{rep} < \text{maxrep} \) then go to Step 2; else go to Step 8.
8. If \( \text{temp} < \text{maxtemp} \) then \{set \( T = \alpha(T) \); go to Step 2\} else \{stop; \( P \) is the approximation to the optimal solution\}.

If the \( \text{succ} \) exceeds \( \text{maxsucc} \), earlier exit is made and the temperature is set to next value.

**B. Annealing**

The function \( \alpha(T) \) in Step 1 and 8 in Algorithm 4 determines the cooling schedule. The rate at which the temperature is reduced is crucial to the success of the annealing process. This is governed by the number of repetitions at each temperature and the rate at which the temperature is reduced. Theory suggests that the system should move close to the stationary distribution of current temperature before temperature reduction, and the temperature should converge gradually to zero. It also suggests that a number of repetitions which are exponential in problem size will be necessary at each temperature \([12]\). Since this implies that SA becomes more complex than the original algorithm using exhaustive search, some heuristic method to reduce the number of iterations is unavoidable. This may be achieved either by using a large number of iterations at each temperature or a small number of iterations at many temperatures. In our experiment, we use the following cooling schedule based on the first approach, which is also widely used in practical applications of SA.

The temperature is reduced by a multiplicative constant, i.e., \( \alpha(T) = aT \) where \( 0 < a < 1 \). Experience has shown that \( a \) in the range of \([0.8, 0.99]\) produces good results. This corresponds to fairly slow temperature reduction. The number of iterations at each temperature may be related to the problem size. In our case, we used \( \text{maxrep} = M - 1 \) as the maximum number of iterations. We also allow early temperature reduction when the number of successful site replacements at each temperature reaches \( \text{maxsucc} = (M - 1)/2 \).

A cooling schedule used in an asymptotic convergence proof, \( \alpha(T) = C/\log(k + 1) \) where \( k \) is the iteration index \([14]\), may be slow because it is so slow that it requires too many iterations for many practical problems. In theory, it is required that the temperature should be allowed to decrease to zero before stopping the algorithm. As \( T \) approaches zero, the probability of acceptance of uphill moves becomes vanishingly small because of the limited precision of computer arithmetic.

In order to avoid wasting time circling around this solution position, the algorithm can be stopped when the number of accepted moves drops below a certain threshold. In our experiment, we employed the following simple rule. **Stopping criterion:** Stop when there are no accepted moves after \( \text{maxrep} \) replacement tries or after \( \text{maxtemp} \) temperature tries. The typical values were \( M - 1 \) for \( \text{maxrep} \) and 10 for \( \text{maxtemp} \). Recall that \( M \) is the GOP size, typically 15.

**VII. Complexity**

Let us consider the computational complexity of (3) assuming that the number of allowable quantization parameters is \( L_q \) and the average number of \( \lambda \) iterations is \( C \lambda \). The complexity of exhaustive search for solving (3), \( C \), is

\[
C = C \lambda 2^{M-1} L_q^M ,
\]

Note that the unit of \( C \) is the number of encoding and decoding needed for one GOP of size \( M \).

Typical values of \( C \lambda \) and \( L_q \) are ten and three for three-level quantizer. Then (17) is about \( 2.3 \times 10^{12} \). Our proposed algorithm using the unimodality in \( N \) and monotonicity for \( Q \) reduces the amount of computation while retaining optimality.

The estimate of the number of iterations for \( N \) is about a half of that of exhaustive search. It is also found in our experiments that the estimate for \( Q \) search is roughly \( M L_q^2 \). The complexity of (3) is then

\[
C = C \lambda 2^{M-2} M L_q^2 ,
\]

which is about \( 1.1 \times 10^7 \). Compared to the exhaustive search, about a \( 10^5 \)-fold computational savings is achieved.

For some applications, even this savings may not be enough. Two heuristics have been introduced mainly to reduce the computation further; one is greedy pruning for \( Q \) search and the other is SA for \((N, P)\) search. Their complexities are roughly on the order of \( M L_q \) and \( M(M - 1) \). The complexity for the whole algorithm is then

\[
C = C \lambda M(M - 1) M L_q = M^2 (M - 1) C \lambda L_q
\]

which is about \( 10^5 \). Compared to (18), another 100-fold savings is obtained. Compared to (17), the savings is about \( 10^7 \)-fold.
VIII. EXPERIMENTS

In our simulation, the quantization parameter is fixed for all macroblocks within each frame and the number of allowable quantization parameters used is three. The common intermediate format (CIF) Tennis and Football sequences were used. The frame rates tested were 30 frames/s for the bit rate 736.5 kb/s. All three color components (Y, U, and V) are encoded. When we used simulated annealing for optimal position search, the number of iterations for given N is \( \frac{M-1}{N} \).

To assess performance based on reasonably limited computation time, instead of results for the optimal scheme, we present the results for the combined heuristic scheme that uses the simulated annealing of Section VI and the greedy pruning suboptimal heuristic of Section V-B. Let us call this scheme SA. The scheme is compared with the conventional nonadaptive scheme which uses four reference (P) frames, which we call the Fixed scheme. The macroblock-level mode decisions (especially, intra mode) are used in both schemes.

Fig. 6 shows SNR results comparing the fixed scheme with SA for Tennis sequence. The average SNR’s are 27.22 dB for SA and 26.86 for Fixed. The corresponding bit rate curve is shown in Fig. 7. The average bit rates are 11.744 Kb/frame for SA and 12.867 Kb/frame for Fixed. Note that SA has 0.36 dB better performance than Fixed even when it is coded with fewer bits by 1.1 Kb/frame. The optimal GOP structure obtained for Tennis has two I frames at frame number 0 and 15 and three P frames at frame numbers 1, 2, and 5. All the rest of the frames are B frames.

The scheme is also tested for the Football sequence. Fig. 8 shows SNR results which compares Fixed with SA. The average SNR’s are 26.88 dB for SA and 26.85 for Fixed. The corresponding bit rate curve is shown in Fig. 9. The average bit rates are 23.576 Kb/frame for SA and 21.696 Kb/frame for Fixed. In this case, the performances can be considered to be similar though the SA scheme spends slightly more bits than the fixed scheme. The optimal GOP structure obtained for Football has two I frames at frame number 0 and 15 and one P frame at frame number 5. All the rest of the frames are B frames.

For Tennis sequence, the decoded images (frame 6) for the SA and the Fixed schemes are compared in Fig. 10 (26.96 dB) and Fig. 11 (26.28 dB). For Football sequence, the decoded images (frame 8) for the SA and the Fixed schemes are compared in Fig. 12 (26.45 dB) and Fig. 13 (24.91 dB). The decoded frames were randomly chosen.
IX. CONCLUSIONS

In this paper, we investigated methods to find optimal reference frame number and positions with fixed bit budget constraint in an operational R-D sense. In the optimal scheme, we exploited the monotonicity and the unimodality of the Lagrangian cost function to reduce the number of R-D data generation. Although the optimal scheme may require more frame memory than the fixed scheme and it may be too complex to be implemented in a practical encoder, it could still be used as a benchmark for performance evaluations of other nonoptimal schemes.

The heuristic schemes that are based on the simulated annealing and the greedy pruning rule can be practical solutions for nonrealtime applications such as CD-ROM and DVD. They also require much less computational complexity and memory than the optimal algorithm does. The algorithms presented here appear to be more promising for nonreal time applications because the large processing delay that is inherent in the algorithms is more tolerable in those applications.

REFERENCES


Jungwoo Lee (S’88–M’94) was born in Seoul, Korea, in 1964. He received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1988 and the M.S.E. and Ph.D. degrees in electrical engineering from Princeton University in 1990 and 1994, respectively.

He has been a Member of Technical Staff working on image processing and video coding at the Information Sciences Research Lab of the David Sarnoff Research Center, Princeton, NJ, since 1994. In 1992, he worked as a summer intern with the Imaging Department of Siemens Corporate Research, Princeton, NJ, working on problems of video storage and retrieval. He has two U.S. patents granted and three more pending in the area of digital video compression and processing. His research interests include image and video coding with emphasis on adaptive algorithms, signal and image processing, and video indexing for multimedia.

Bradley W. Dickinson (S’70–M’74–SM’82–F’87) was born in St. Marys, PA, on April 28, 1948. He received the B.S. degree in engineering from Case Western Reserve University, Cleveland, OH, in 1970 and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1971 and 1974, respectively.

Since 1974, he has been a faculty member at Princeton University, Princeton, NJ, where he is currently Professor of Electrical Engineering. His research interests include signal and video processing, system theory, and neural networks.

Dr. Dickinson is a past Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and of the IEEE TRANSACTIONS ON INFORMATION THEORY. He has served on the Board of Governors of the Information Theory Society, as a Technical Committee Chairman for the Control Systems Society, as a member of the Multidimensional Signal Processing Technical Committee for the Acoustics, Speech, and Signal Processing Society, and as member of the Neural Networks Council. He is a founding Managing Editor of the journal Mathematics of Control, Signals, and Systems (Springer-Verlag). He was a founding Co-Editor of the E-LETTER on Systems, Control, and Signal Processing. He is author of the book Systems: Analysis, Design, and Computation (Prentice-Hall, 1992). He is a member of Tau Beta Pi.