

 Open access • Journal Article • DOI:10.1109/TIP.2009.2014259

Consistent Picture Quality Control Strategy for Dependent Video Coding

— [Source link](#) 

Kao-Lung Huang, Hsueh-Ming Hang

Institutions: National Chiao Tung University

Published on: 01 May 2009 - IEEE Transactions on Image Processing (IEEE)

Topics: Subjective video quality, Image quality, Inter frame, High-motion and Rate–distortion theory

Related papers:

- [Rate control of MPEG video for consistent picture quality](#)
- [MPEG motion picture coding with long-term constraint on distortion variation](#)
- [A sequence-based rate control framework for consistent quality real-time video](#)
- [Overview of the H.264/AVC video coding standard](#)
- [Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/consistent-picture-quality-control-strategy-for-dependent-1e9y7o6unc>

Consistent Picture Quality Control Strategy for Dependent Video Coding

Kao-Lung Huang and Hsueh-Ming Hang, *Fellow, IEEE*

Abstract—Typically, a video rate control algorithm minimizes the average distortion (denoted as MINAVE) at the cost of large temporal quality variation, especially for videos with high motion and frequent scene changes. To alleviate the negative effect on subjective video quality, another criterion that restricts a small amount of quality variation among adjacent frames is preferred for practical applications. As pointed out by [20], although some existing proposals can produce consistent quality videos, they often fail to fully utilize the available bits to minimize the global total distortion. In this paper, we would like to achieve the triple goal of consistent quality video, minimizing the total distortion, and meeting the bit budget strictly all at the same time on the interframe dependent coding structure. Two approaches are taken to accomplish this goal. In the first algorithm, a trellis-based framework is proposed. One of our contributions is to derive an equivalent condition between the distortion minimization problem and the budget minimization problem. Second, our trellis state (tree node) is defined in terms of distortion, which facilitates the consistent quality control. Third, by adjusting one key parameter in our algorithm, a solution in between the MINAVE and the constant quality criteria can be obtained. The second approach is to combine the Lagrange multipliers method together with the consistent quality control. The PSNR performance is degraded slightly but the computational complexity is significantly reduced. Simulation results show that both our approaches produce a much smaller PSNR variation at a slight average PSNR loss as compared to the MPEG JM rate control. When they are compared to the other consistent quality proposals, only the proposed algorithms can strictly meet the target bit budget requirement (no more, no less) and produce the largest average PSNR at a small PSNR variation.

Index Terms—Bit allocation, consistent quality control, H.264, quality smoothing, video rate-distortion control.

I. INTRODUCTION

VIDEO coding technologies have been progressing very fast in the past two decades. The MPEG-4 AVC/H.264 is the latest international video coding standard [1], [2]. To achieve the optimal rate-distortion (R-D) performance on a

Manuscript received April 08, 2008; revised December 03, 2008. First published March 24, 2009; current version published April 10, 2009. This work was supported in part by the National Science Council (Taiwan, R.O.C.) under Grant NSC 91-2219-E009-011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Srdjan Stankovic.

K.-L. Huang is with the Department of Electrical Engineering, National Chiao-Tung University, Hsinchu, 30010, Taiwan, R.O.C. (e-mail: hcting@seed.net.tw).

H.-M. Hang is with Department of Electrical Engineering, National Chiao-Tung University, Hsinchu, 30010, Taiwan, R.O.C., and also with the Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10061, Taiwan, R.O.C. (e-mail: hmhang@mail.ntu.edu.tw).

Digital Object Identifier 10.1109/TIP.2009.2014259

specific coding structure, the so-called *rate control* algorithms are proposed to determine the best quantization parameter (QP) for a coding unit (which can be a macroblock (MB) or a frame) and these algorithms should also prevent the buffer(s) from underflow or overflow in the environment of a constant bit rate (CBR) channel or a variable bit rate (VBR) channel. In this paper, we will focus on the frame-level bit rate control over a CBR channel.

Many types of rate control algorithms have been developed [3]. The rate control problem in video coding becomes highly complicated mainly due to the motion estimator and other interframe operations in the system. The parameters selected in encoding the current frame affect the parameter selection in the subsequent frames, which results in the so-called *dependent coding* structure [4]. Two approaches are taken to solve this problem. A sub-optimal approach simplifies the original formulation by adopting the *independent coding* structure, which picks up the best parameters for the current frame without considering their effects on the future frames, for example, the method in [5]. Many practical *one-pass* or *two-pass* algorithms belong to this category and they include R-D models such as the classical statistical model [6], [7], the quadratic model [8], and the rho-domain model [9]. The R-D optimality is not guaranteed in this approach because of the unavailability of future frames. The other approach is to reduce the complexity of the parameter search so that a near-optimal solution can be obtained at a reasonable computational cost. Mathematically, the exhaustive search for the optimal frame QPs in a group of pictures (GOP) is equivalent to finding the optimal path in a tree. Potentially, this approach can identify the globally optimal solution, which cannot be accomplished by the first sub-optimal approach. However, the tree grows exponentially as more pictures are coded. Therefore, several methods are proposed to reduce the search complexity, for example, the monotonicity assumption [10], the node clustering [11], the steepest descent search [12], and the interframe R-D model [13]. The first two methods are adopted in this paper and will be elaborated in Section III.

There are two commonly used optimization criteria in designing a rate-control algorithm: minimum average distortion (the MINAVE criterion) and minimum maximum distortion (the MINMAX criterion). The MINAVE criterion [14] aims at minimizing the total distortion under a given bit budget. This optimal goal is widely adopted and is well studied in the literature. Examples are the algorithms mentioned in the previous paragraph. However, this MINAVE goal is attained often at the expense of a possibly larger frame-to-frame quality variation. From the perspective of human visual system (HVS), a video

sequence with nearly constant quality or *consistent quality* is more desirable [15]. Therefore, the MINMAX criterion [14] is proposed to minimize the maximal distortion for a given bit budget. Coupled with the dependent coding structure, achieving the MINMAX target becomes a very complicated issue. Several methods have been proposed such as dynamic programming [14], lexicographic algorithm [16], low-pass filtering of rate-distortion functions [17], minimum distortion variation [13], [18] and an iterative frame bit allocation algorithm [19]. Often, the final results produced by these proposals do not achieve strictly the global MINMAX target. They typically produce videos with a slowly varying quality, or in other words, with a consistent quality, and this is practically all we need.

Extensions of the one-pass algorithms are frequently adopted by the researchers in the last paragraph. However, their assumption of similar image statistics in near-by frames does not hold for videos with high motion and scene changes. Also, the one-pass approach may fail to achieve consistent quality on fast motion sequences due to limited bit budget and unavailability of future frames [18]. Furthermore, these methods often decrease the frame-to-frame quality variation without paying attention to the total distortion. Therefore, a hybrid MINMAX/MINAVE method was suggested to increase the overall quality after finding the MINMAX solution [20].

In this paper, we tackle the dependent MINAVE and consistent-quality problems simultaneously. More specifically, we would like to achieve the consistent quality goal across the entire sequence and, in the meantime, to meet the target bit rate accurately and to minimize the total distortion. The tradeoff between average distortion and consistent quality is controlled by one key parameter, namely, the maximal quality variation constraint. One method to solve the above optimization problem with finite parameter set is the dynamic programming approach. By adopting the monotonicity and clustering concepts, the tree structure in the *dependent video coding* is converted into a trellis diagram. Thus, the Viterbi algorithm can be employed to find the truly optimal solution in this dependent coding problem. The trellis state (tree node) is defined in terms of distortion to facilitate the consistent quality control. In addition, a fast technique is proposed to decrease the computation in the branch expansion process. By adjusting the key parameters in our scheme such as cluster size, we can decrease the computational complexity at the cost of minor performance loss.

A second method is proposed based on the Lagrange multipliers. To ensure the global optimality on the dependent coding platform, an iterative scheme is designed to find the best lambda (λ) parameter (Lagrange multiplier) in the Lagrange cost or Lagrangian. This algorithm backtracks many times to narrow down a valid λ range containing the optimal λ . Then, the best λ value is identified by a fast search algorithm. This scheme runs much faster than the trellis-based approach. Its performance is close to but slight lower than that of the trellis-based approach.

Despite the optimality of both methods suggested in this paper, their real-time implementation is still beyond the current hardware capability. Thus, the proposed algorithms may be more suitable for off-line applications such as DVD playback when video quality is the major concern. We implement our algorithm on the new and very efficient H.264 coder and evaluate its performance.

This paper is organized as follows. In Section II, we introduce the rate-control problem in video coding and derive an equivalent condition between the distortion minimization problem and the budget minimization problem. Two proposed algorithms are described in Section III: a) the trellis-based algorithm with Viterbi search and b) a Lagrangian-based iterative algorithm with bisection search. Section IV presents the simulation results to show the effectiveness of our algorithm. These results are compared with existing MINAVE and MINMAX schemes. Also, the effect of control parameters on PSNR and complexity is studied. Section V summarizes the findings and their limitations in this paper.

II. PROBLEM FORMULATION AND DISTORTION-RATE FUNCTION

The frame-level bit allocation problem and the uniqueness property of the distortion-rate function are described in this section. In our selected structure, we encode a frame and all its macroblocks using the same QP. The notion of quality in this paper is the well adopted image objective criterion, PSNR.

A. Dependent MINAVE Bit Allocation Problem

In the (forward prediction) dependent coding formulation, the k th frame distortion and bits, i.e., d_k and r_k , depend on the current and previous frame QP values. Let $Q = \{0, 1, \dots, 51\}$ be the set of quantization parameter values in the H.264 standard video codec. Given K frames and a total bit budget R_T , our goal is to minimize the overall distortion D by choosing the optimal frame-level QP values $\mathbf{q}^* = \{q_0^*, q_1^*, q_2^*, \dots, q_{K-1}^*\}$ for all K frames, where $q_k^* \in Q$, for $k = 0, 1, \dots, K - 1$. That is

$$\mathbf{q}^* = \arg \min_{q_k \in Q} D = \sum_{k=0}^{K-1} d_k(q_0, \dots, q_k)$$

subject to the constraints

$$R = \sum_{k=0}^{K-1} r_k(q_0, \dots, q_k) \leq R_T \text{ and} \\ |f(d_k) - f(\bar{D})| \leq \delta_P, \quad \forall k \in \{0, 1, \dots, K - 1\} \quad (1)$$

where \bar{D} is the average distortion for all K frames and $f(\cdot)$ is the PSNR function calculated by $f(d) = 10 \log_{10}(255^2 \times \text{FPN}/d)$, where FPN is the pixel number in a frame. The second constraint in (1) is added to achieve the consistent quality video; that is, the difference between the frame PSNR and the average sequence PSNR is limited by $\pm \delta_P$.

Another important function of a rate control algorithm is to avoid the buffer underflow and overflow problems. The MPEG standard imposes a hypothetical decoder model on a legal bit stream, namely, Video Buffer Verifier (VBV). There are three prescribed operation modes in VBV. In this study, we consider only the constant bit rate (CBR) mode; i.e., the channel rate is constant. We assume that the decoder buffer is large enough to eliminate the buffer overflow problem. In more details, the buffer is initially empty. To avoid the buffer underflow problem, bits in the decoder buffer accumulate for a specific time before the bits of the first frame are removed. Afterwards, the decoder buffer continues receiving constant-rate bits from the channel

and the decoder removes the bits in buffer at regular frame-time intervals. Essentially, the buffer underflow problem imposes a delay T_d on the decoder. For frame k , the buffer occupancy is

$$B_k = B_0 + kc - \sum_{i=0}^k r_i = T_d c + kc - \sum_{i=0}^k r_i \quad (2)$$

where B_0 is the initial buffer occupancy and c = channel bit rate/frame rate. The buffer underflow is avoided if the constraint $B_k \geq 0$ for all $k \geq 0$ is satisfied. In other words, the decoder delay T_d shall be selected to ensure that the buffer contains at least r_k bits, when the decoder starts decoding frame k for all $k \geq 0$.

B. Uniqueness of Distortion-Rate Function

In the conventional MINAVE problem, we minimize the total distortion subject to a given bit constraint. However, in order to achieve a consistent quality video, it is more convenient if the distortion, not the bit rate, is the controllable argument in our process. That is, we prefer $MinR(D_T)$ rather than $MinD(R_T)$.

In the classical information theory, the distortion-rate function, $D(R)$, is a nonincreasing, convex function and its slope must be both nonpositive and nondecreasing. Then, the rate-distortion function, $R(D)$, the inverse of $D(R)$, is a legal nonincreasing, convex function too. As a result, the solution to the $MinR(D_T)$ problem is identical to that to the $MinD(R_T)$ problem. However, these ideal properties of the rate-distortion function may not true for the real-data case. Therefore, we study the relation of these two solutions in the operational sense and derive the proposition as follows.

Proposition: Given a rate-distortion coder with control parameters of discrete and finite values, we consider the operational $R(D)$ and $D(R)$ functions. In other words, $D(R)$ is the achievable distortion for the given bit rate R . $R(D)$ is similarly defined. Then, the optimal solution, (D_0^*, R_0) , to the minimum distortion problem, i.e., $MinD(R_T)$, is also the optimal solution, (D_1, R_1^*) , to the minimum budget problem, i.e., $MinR(D_T = D_0^*)$, if the optimal distortion function $D^*(R)$ is a one-to-one mapping, where (D^*, R) is the solution set to the $MinD(R_T)$ problem at the given R_T budget bits.

Proof: Since (D_1, R_1^*) is the optimal solution to $MinR(D_T = D_0^*)$, it implies $D_1 \leq D_0^*$. On the other hand, D_0^* is the optimal solution (least amount of distortion) to $MinD(R_T)$, thus $D_1 \geq D_0^*$. Consequently, we have $D_1 = D_0^* = D_T$. The optimal solution (D_0^*, R_0) of $MinD(R_T)$ implies $R_0 \leq R_T$. In addition, R_1^* is the optimal solution (least amount of bits) of $MinR(D_T = D_0^*)$; it thus implies $R_1^* \leq R_0$. Consequently, we have $R_1^* \leq R_0 \leq R_T$. Now, if $D^*(R)$ is a one-to-one function, the relation $R_1^* = R_0$ must be true because $D_1 = D_0^*$. Therefore, the solutions to these two problems, $MinR(D_T = D_0^*)$ and $MinD(R_T)$, are identical if $D^*(R)$ is a one-to-one function. (QED)

III. CONSISTENT QUALITY CONTROL ALGORITHM

Two approaches are chosen to solve the interframe dependent coding problem in this study. We start with the trellis-based

approach. First, the tree structure inherent in dependent coding is reduced to the trellis structure. Then, the branch expansion process is described and the Viterbi search is used to solve the bit allocation problem. Next, a fast branch expansion algorithm extended from a previous proposal is presented. In the last sub-section, we propose the Lagrange multipliers approach. An iterative structure is designed for finding the optimal lambda value in the Lagrange cost. To speed up this iterative process, a couple of the existing but independently proposed fast schemes are included with proper modifications.

A. Trellis Representation of the Tree Structure

In the dependent coding structure, the current frame distortion and bits depend not only on the current QP but also on the previous frame QPs. Given 52 possible QP values, there are 52 possible coded pictures (each coded using a different QP value) for the first frame. Each coded picture is associated with a (distortion, rate) pair after coding. Each of them leads to 52 possible second-frame pictures. Therefore, there are in total 52^2 possible pictures (or states, nodes in a tree) for the second picture. The picture (or state) number grows exponentially as more frames are coded. All the possible picture sequences thus form a tree structure. The computational complexity of finding the optimal solution in a tree becomes a serious issue.

Two approaches were suggested to reduce the growing number of states. The state pruning technique was proposed by [10] and the state clustering approach was proposed by [11]. In the first approach [10], the state in a tree is denoted by the accumulated frame coded bits. The theoretical basis of state pruning is the ‘‘monotonicity’’ assumption that a better current coding frame will lead to a more efficient coding in the future [10]. Although this monotonicity condition is not always guaranteed as pointed out by [21], our experimental results indicate this assumption is typically true. Therefore, the Markovian condition (the future optimal path depends only on the current state not the previous one) is created. The Viterbi Algorithm (VA) can thus be applied. As a result, when multiple branches arrive at the same state, only one branch of the minimal accumulated distortion is selected as the survivor and the complexity is largely reduced. The second complexity reduction approach adopts the notion of ‘‘cluster’’ [11], which merges a few neighboring nodes (states) into one cluster because these nodes (in one cluster) have similar states (buffer level in [11]) and thus lead to similar final results.

We adopt both the concepts of monotonicity and cluster in this study. However, for the quality variation control purpose in this study, the distortion value (represented by PSNR) is used as the state variable. In addition, because the PSNR value is a real number, the problem of infinite states occurs in this formulation. Therefore, a cluster representing a distinct range of PSNR values is defined as a *state*. The *cluster size* parameter P_Δ is used to define the span of a cluster. To convert the tree structure into a trellis, it is necessary to restrict the dynamic range of admissible PSNR. It is set by the lowest quality, denoted by P_L , and the highest quality, denoted by P_U . This range should include all the PSNR values in the optimal solution and is chosen empirically. Consequently, the number of states equals to $1 + \lfloor (P_U - P_L) / P_\Delta \rfloor$, where $\lfloor x \rfloor$ denotes the integer part of

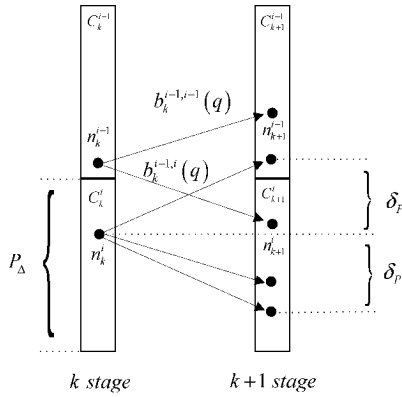


Fig. 1. Illustration of cluster, node, branch, P_Δ , and δ_P definitions.

x . Because there are only a finite number of states, the tree structure is degenerated into the trellis structure. In contrast, the concept of cluster is proposed to reduce the tree search complexity in [11] and now is extended for the purposes of both defining finite states and reducing complexity in this study.

The rest is the detailed description of our trellis structure. Fig. 1 illustrates the relation among cluster, node, branch, P_Δ , and δ_P .

- **Cluster:** The notation $c_k^i(R_k)$ represents a cluster with index i at stage (frame) k , where $i \in [0, \lfloor (P_U - P_L)/P_\Delta \rfloor]$ and $k \in [0, K - 1]$. The i th cluster PSNR range is $[P_L + iP_\Delta, P_L + (i + 1)P_\Delta)$. A cluster may contain a number of nodes in it. The best performing node (in the R-D sense) inside a cluster is chosen to be the *representative node* of this cluster.
- **Node:** A node $n_k^i(D_k, R_k)$ represents a legal operating point of the coding result, whose PSNR value is in the cluster i at frame k , where $i = \lfloor (f(\bar{D}_k) - P_L)/P_\Delta \rfloor$, and $\bar{D}_k = D_k/k$. D_k and R_k are the accumulated coded distortion and bits before encoding frame k , respectively.
- **Branch:** A branch connects two nodes in the trellis diagram. The notation $b_k^{i,j}(q_k)$ indicates that it stems from the representative node in cluster i at frame k and it ends at a node in cluster j at frame $k + 1$. It uses q_k to quantize frame k . It produces a next stage node $n_{k+1}^j(D_{k+1} = D_k + d_k^j(q_k), R_{k+1} = R_k + r_k^j(q_k))$, where $j = \lfloor (f(\bar{D}_{k+1}) - P_L)/P_\Delta \rfloor$ and $\bar{D}_{k+1} = D_{k+1}/(k + 1)$, if the three conditions, $R_{k+1} \leq R_T$, $|f(d_k) - f(\bar{D}_k)| \leq \delta_P$ and $P_L \leq f(\bar{D}_{k+1}) \leq P_U$, are all satisfied. A rate-distortion pair $(r_k^i(q_k), d_k^i(q_k))$ is associated with this branch. Note that the average sequence PSNR value $f(\bar{D})$ is not available until the end of the encoding process. It is thus approximated by the current $f(\bar{D}_k)$ value.

B. Branch Expansion and Frame-Level Bit Allocation

Let two nodes of n_{k-1}^j and n_k^i be connected by a branch $b_{k-1}^{i,j}(q_{k-1})$. In the branch expansion process for node n_k^i , all the QPs satisfying the following three constraints are examined (that is, they are used to quantize data in frame k): a) PSNR range $P_L \leq f(\bar{D}_{k+1}) \leq P_U$, b) bit budget $R_{k+1} \leq R_T$, and c) quality variation $|f(d_k) - f(\bar{D}_k)| \leq \delta_P$. The previous frame QP value,

q_{k-1} , is selected to be the center QP value, denoted by QP_c , and the examined QPs are expanded from the center value gradually by $q_{k,n} = q_{k-1} \pm n$, where the step index n is incremented by one until any of the above constraints is violated.

The first frame (I frame) in a sequence is by default the first active node. In the following frames (P frames), the number of branches and nodes grows exponentially if they are not eliminated or merged. The adaptation of the cluster concept allows the merge of nodes with similar distortion. A cluster containing at least one expanding node in it is called *active cluster*. When a small cluster size, say, $P_\Delta \approx 0.1$ dB, is in use, typically only the branch of least accumulated bits and its associated node will be the single survivor in this cluster. The survivor node in an active cluster is defined as an *active node*. The ‘‘monotonicity assumption’’ enables the elimination of weaker branches (branches with higher bit rates) ending at the same node (cluster). That is, in the backtracking process, only the active node with the smallest total distortion and permissible bit usage is selected. Therefore, the goal of minimizing the total distortion is achieved.

To accomplish the consistent quality video goal, $\delta_P \leq 0.4$ dB is usually adopted. Overall, the proposed quality control algorithm is summarized below.

Algorithm 1: Trellis-Based Consistent Quality Control (TCQC) Algorithm

- Step 1: Initialize the values of R_T , δ_P , P_L , P_U , and P_Δ .
- Step 2: Encode the first I frame using all quantization values. Prune the branches that violate any of the two constraints: the PSNR range $P_L \leq f(\bar{D}_k) \leq P_U$ and the bit budget $R_k \leq R_T$.
- Step 3: If multiple branches merge at the same destination cluster, select the branch with the least accumulated bits and its corresponding node becomes the survivor. At the end of this step, each cluster contains only one active node, which is connected to only one surviving branch. Save the context information of the survivor nodes.
- Step 4: Expand all active nodes for the next I- or P-frame. Encode the next frame (frame k) using all allowable quantization scales. Prune the branches which violate any of the three constraints: the PSNR range $P_L \leq f(\bar{D}_{k+1}) \leq P_U$, $R_{k+1} \leq R_T$, and the quality variation constraint $|f(d_k) - f(\bar{D}_k)| \leq \delta_P$.
- Step 5: If the current frame is not the last frame in the sequence, go to Step 3.

Otherwise, among all active clusters, choose the survivor node with the best overall quality as the final solution. Backtrack along the optimal path connecting to the starting frame of this sequence. We thus obtain the optimal frame-level QP and bits for each frame. This sequence is then done.

C. Fast Branch Expansion Process

Generally, a complete video encoding process is executed whenever a branch is expanded. In the MPEG JM reference software, the coding parameter selection is done by two components: the rate-control algorithm and the rate distortion optimization (RDO) process. This 2-stage coder control structure is well recognized for its efficiency for a highly complicated hybrid video coder such as H.264. But the RDO process is costly in computation. The RDO process needs a QP input value for its operation and it outputs the coding modes, distortion, header bits, and residual signals. On the other hand, a typical rate-control algorithm needs the modes etc. information to pick up the best QP for quantizing the current MB or frame. Therefore, these two components depend on each other for supplying their inputs, a chicken and egg problem [22]. Let QP_1 and QP_2 denote the QPs used by the RDO process and the quantization process, respectively. The initial value of QP_1 is generally not equal to the value of QP_2 . Therefore, an iterative procedure has been proposed for updating QP_1 (for example, $QP_{1\text{new}} = (QP_1 + QP_2/2)$) after the first set of QP_1 and QP_2 are obtained [22].

It is reported that the coding PSNR loss is less than 0.2 dB when $|QP_1 - QP_2| \leq 3$ [22]. When a frame is encoded twice using two sets of QP values, namely, $QP_1 = QP_2 = q_1$ and $QP_1 = QP_2 = q_2$, separately, we run RDO only once with $QP_1 = q_1$. Using the aforementioned property, the same RDO outputs are used for quantization in both cases, $QP_2 = q_1$ and $QP_2 = q_2$, if q_1 and q_2 are sufficiently close. We thus save one RDO computation.

To lower the approximation error, we restrict the approximation range by $|QP_1 - QP_2| \leq 2$. The fast branch expansion process now runs as follows. First, the current frame is encoded using the center QP defined in Section III-B, i.e., $QP_1 = QP_2 = QP_c$. Then, the upper and lower two branch expansions can be easily generated by performing the quantization processes four times with $QP_2 = QP_c \pm 1$, $QP_2 = QP_c \pm 2$. As a result, five branch expansions are generated at the cost of computing one RDO process and five quantization processes. If more branch expansions are needed, another complete video encoding process is needed, for example, $QP_1 = QP_2 = QP_c + 5$ or $QP_c - 5$. Finally, to prevent the approximation error propagation to the next stage, a complete video encoding process, i.e., running RDO and quantization with the chosen final QP, is executed again for each active cluster.

D. Technique Based on the Lagrange Multipliers

Another optimization technique, the so-called *Lagrange multipliers method* can also be used to find the optimal operation point on the rate-distortion curve [11]. We define the Lagrange cost to be $J(\lambda) = \sum_k d_k + \lambda r_k = \sum_k J_k(\lambda)$. The goal becomes

$$\begin{aligned} \text{Min}_{\lambda \in Q^k} J(\lambda) &= \text{Min}_{q_k \in Q} \sum_{k=0}^{K-1} (d_k + \lambda r_k) \\ &= \sum_{k=0}^{K-1} \text{Min}_{q_k \in Q} (d_k + \lambda r_k). \end{aligned} \quad (3)$$

It is well-known that the optimal solution to the minimizing distortion problem with budget constraint, denoted by $\text{Min}D(R_T)$, is equivalent to that of minimizing the Lagrange cost, $\text{Min}J(\lambda^*)$ in (3) with $R(\lambda^*) = R_T$ [11]. The key step in finding the optimal solution is to identify λ^* , the optimal value of lambda. In general, this optimal λ^* solution can be iteratively solved [11]. However, in this study, we impose two additional constraints: the consistent quality constraint, $|f(d) - f(\bar{D})| \leq \delta_P$ and the PSNR range constraint, $P_L \leq f(\bar{D}) \leq P_U$. We develop an iterative process to solve this new and more complex problem as follows.

First, the budget constraint of $R(\lambda^*) = R_T$ is relaxed. We intend to find a proper lambda range, denoted by $[\lambda_{LL}, \lambda_{UU}]$, such that the solution to the $\text{Min}J(\lambda)$ problem with a lambda located inside this range shall satisfy all three constraints, $P_L \leq f(\bar{D}) \leq P_U$, $|f(d) - f(\bar{D})| \leq \delta_P$, and $R(\lambda_{UU}) < R_T < R(\lambda_{LL})$. Therefore, the optimal lambda value λ^* is guaranteed to locate in the selected range. Next, a fast bisection algorithm in [23] is employed to find the solution to the $R(\lambda^*) = R_T$ problem. That is, the lambda search process iterates until the predefined bit rate tolerance, i.e., $R_T - R(\lambda)/R_T \leq \varepsilon$, is satisfied. And the (optimal) QPs are a byproduct in this process.

In the following, we describe how the constraints are satisfied in the aforementioned process of finding the lambda range. For a given frame, we examine only the valid QP values that satisfy the quality constraint, $|f(d) - f(\bar{D})| \leq \delta_P$. The picture coding process is similar to the fast branch expanding step described earlier. To satisfy the other two constraints $P_L \leq f(\bar{D}) \leq P_U$ and $R(\lambda_{UU}) < R_T < R(\lambda_{LL})$, we start with two initial lambda values, λ_U and λ_L , such that both $R(\lambda_U) < R_T < R(\lambda_L)$ and $[P_L, P_U] \subset [P(\lambda_U), P(\lambda_L)]$ are satisfied. Then, the center value in the current lambda interval is used as the *test lambda* λ_T to determine whether the solution to the $\text{Min}J(\lambda_T)$ problem satisfies the constraint $P_L \leq f(\bar{D}(\lambda_T)) \leq P_U$. If the current average PSNR is lower than P_L , a smaller lambda should be used, and, thus, the lower subinterval $[\lambda_L, \lambda_T]$ is selected as the lambda interval for the next iteration. Equivalently, the test lambda value is decreased in the next iteration. On the other hand, if the current average PSNR is larger than P_U , the upper subinterval $[\lambda_L, \lambda_T]$ is selected as the lambda interval for the next iteration, which increases the test lambda value in the next iteration. We check the average PSNR value whenever a frame is coded. If either of the above conditions happens, we need to re-encode the video sequence from the first frame again using the new lambda range. This process continues until the chosen λ_T leads to a successful coding of the entire video sequence. At the end, if the resulting bits are smaller than the bit budget, the latest test lambda value is referred as λ_{UU} . The same process is performed in the lambda interval $[\lambda_T, \lambda_{UU}]$ to obtain λ_{LL} value, but note that the obtained λ_{LL} value shall satisfy $R(\lambda_{LL}) > R_T$. Theoretically, if the values of P_L, P_U, λ_L , and λ_U are properly selected (so that the optimal solution exists), because the R-D curve is convex, this algorithm converges. Overall, the iterative lambda optimization steps are summarized below.

Algorithm 2: Lagrangian-based Consistent Quality Control (LCQC) Algorithm

Step0: Start with two values λ_U and λ_L such that $R(\lambda_U) < R_T < R(\lambda_L)$ and $[P_L, P_U] \subset [P(\lambda_U), P(\lambda_L)]$. Set $\lambda_T = (\lambda_L + \lambda_U)/2$ and frame index $k = 0$.

Step 1: Given λ_T , use the fast branch expansion technique to examine all the QPs that satisfy $|f(d_k) - f(\bar{D}_k)| \leq \delta_p$.

Step 2: If $P_L \leq f(\bar{D}(\lambda_T)) \leq P_U$, go to Step 3.

Step 3: Else if $f(\bar{D}(\lambda_T)) > P_U$, set $\lambda_L = \lambda_T$; otherwise ($f(\bar{D}(\lambda_T)) < P_L$), set $\lambda_U = \lambda_T$. Let $\lambda_T = (\lambda_L + \lambda_U)/2, k = 0$ (start from the first frame again), go to Step 1.

Step 3: Encode the current frame again using the up-to-date QP value. If the current frame is not the last frame in the sequence, let $k = k + 1$, go to Step 1.

Step 4: If $R(\lambda_T) < R_T$ set $\lambda_{UU} = \lambda_U = \lambda_T$. Else set $\lambda_{LL} = \lambda_L = \lambda_T$. If the lambda interval boundaries, λ_{LL} and λ_{UU} , are both found, go to Step 5. Else let $\lambda_T = (\lambda_L + \lambda_U)/2, k = 0$, go to Step 1.

Step 5: Perform the fast bisection search algorithm [23] in the lambda range $[\lambda_{LL}, \lambda_{UU}]$ to find the optimal λ^* , i.e., $R(\lambda^*) = R_T$. The usual stop rule $R_T - R(\lambda)/R_T \leq \varepsilon$ is adopted. A few assistant formulas are proposed in [23] so that this search process converges rather fast. Normally, this step takes 2 to 4 recursions. The final λ^* and its associated QPs are our optimal solution to (1).

Typically, Steps 1 and 3 require only one branch expansion process (to examine the valid QPs) and one complete encoding process (to prevent approximation error propagation), respectively. The computational complexity mainly comes from the number of iterations. It usually takes 5 to 8 iterations to complete this lambda search. Detailed simulation results including PSNR and computing time are discussed in Section IV.

IV. SIMULATION RESULTS

We have implemented the proposed quality control algorithm on MPEG-4 AVC/H.264 video coder with the rate-distortion optimization (RDO) option turned on. Experiments are performed using the standard MPEG video sequences, *Foreman*, *Table Tennis*, *News*, and *Stefan*. All test videos are 300 frames in QCIF size. The GOP size is 30. Only I- and P- frames are in use. The PSNR range in each case is estimated from the minimum PSNR and the maximum PSNR obtained by applying JM 7.6 to the test video sequence. Simulations are performed on a 3-GHz Intel Pentium CPU.

We conduct four sets of simulations to evaluate performance of the proposed TCQC and LCQC algorithms. In the first experiment, the TCQC algorithm is tested at different bit rates to show its effectiveness on bit allocation, as compared with the JM and the constant QP schemes. The JM7.6 rate control scheme is unable to select a QP for the first frame. For fair comparison,

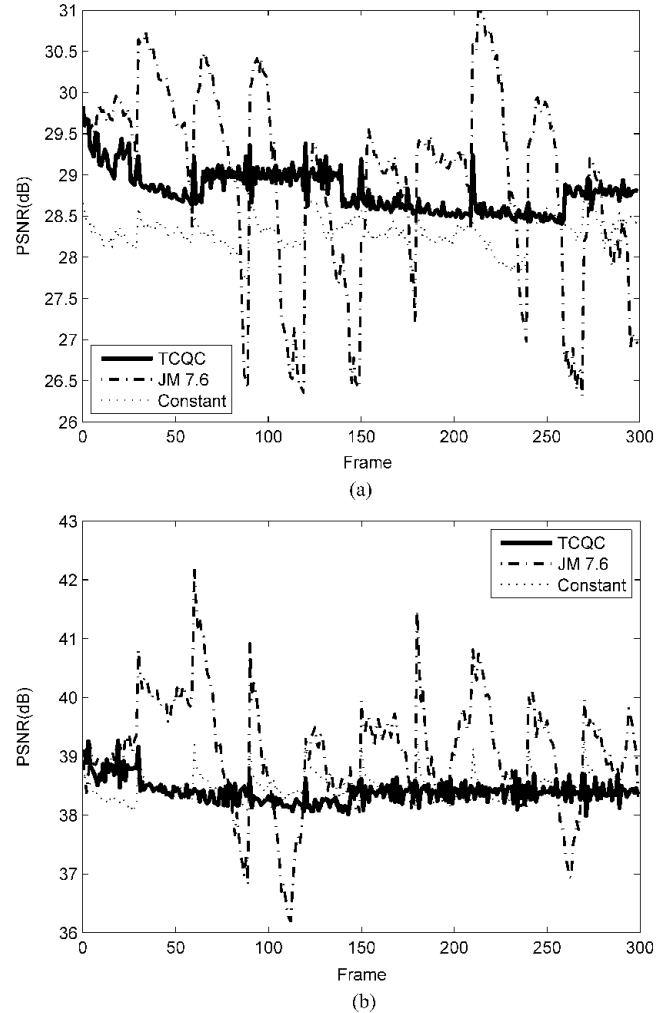


Fig. 2. PSNR plots of the TCQC, JM 7.6, and Constant QP algorithms for News at two bit rates. (a) 24 kbps. (b) 112 kbps.

the first QP is set to be identical to that of the TCQC algorithm. Also, the best constant QP case is shown, which is produced by using a single QP value for the entire sequence. In this experiment, all possible QP values are tried and the one which produces bits closest to the target bits is chosen. Next, the PSNR and complexity of the LCQC algorithm are compared with two published algorithms, LPF in [17] and MultiStage in [19]. In the third and fourth experiments, TCQC and LCQC are compared. Several δ_p and P_Δ values are tested to show the PSNR and complexity tradeoff.

A. Performance Comparison With Constant QP and JM

The TCQC algorithm is evaluated on four different video sequences at three different bit rates to show its effectiveness on bit allocation. The *Foreman* sequence contains mainly a talking head with a scene change near the end, the *News* sequence contains some amount of background changes, the *Table Tennis* sequence has a scene change in the middle, and the *Stefan* sequence has high motion. Two other schemes, namely, JM 7.6 and constant QP, are also applied to these sequences. The parameters used in this experiment are the cluster size $P_\Delta = 0.1$ dB and the maximal quality variation $\delta_p = 0.4$ dB. The PSNR

TABLE I
COMPARISONS OF MINIMUM PSNR, MAXIMUM PSNR, AVERAGE PSNR, PSNR VARIANCE, BIT RATE, AND DECODING DELAY FOR JM 7.6, TCQC, AND CONSTANT QP SCHEMES ON THE FOREMAN SEQUENCE

Target Rate	Method	Rate(% error)	PSNR(dB)				Delay(s)
			Min	Max	Avg	Var	
24 kbps	JM7.6	24.05 (+0.20%)	22.53	30.37	26.35	3.32	0.43
	TCQC	23.83 (-0.73%)	25.62	27.15	26.11	0.07	0.73
	Fixed QP=46	22.11 (-7.86%)	24.42	28.70	25.81	0.49	0.20
64 kbps	JM 7.6	64.10 (+0.15%)	25.46	35.35	32.30	2.99	0.30
	TCQC	63.38 (-0.97%)	31.76	33.09	32.30	0.04	0.67
	Fixed QP=36	58.50 (-8.59%)	30.24	33.79	31.91	0.77	0.40
112 kbps	JM7.6	112.04 (+0.04%)	30.60	38.46	35.43	2.26	0.33
	TCQC	111.46 (-0.48%)	34.92	36.47	35.38	0.06	0.40
	Fixed QP=31	104.97 (-6.28%)	33.15	36.79	35.09	0.97	0.50

TABLE II
COMPARISONS OF MINIMUM PSNR, MAXIMUM PSNR, AVERAGE PSNR, PSNR VARIANCE, BIT RATE, AND DECODING DELAY FOR JM 7.6, TCQC, AND CONSTANT QP SCHEMES ON THE NEWS SEQUENCE

Target Rate	Method	Rate(% error)	PSNR(dB)				Delay(s)
			Min	Max	Avg	Var	
24 kbps	JM7.6	24.19 (+0.79%)	26.31	30.99	28.98	1.36	0.57
	TCQC	23.95 (-0.20%)	28.4	29.83	28.81	0.05	0.43
	Fixed QP=42	22.25 (-7.28%)	27.71	28.91	28.30	0.04	0.33
64 kbps	JM7.6	64.23 (+0.35%)	32.57	38.44	35.46	1.07	0.47
	TCQC	62.95 (-1.65%)	34.39	35.76	34.81	0.04	0.37
	Fixed QP=32	61.62 (-3.72%)	34.39	35.69	34.91	0.06	0.33
112 kbps	JM7.6	112.31 (+0.28%)	36.20	42.20	39.06	0.92	0.37
	TCQC	109.71 (-2.05%)	37.99	39.27	38.41	0.04	0.30
	Fixed QP=27	107.25 (-4.24%)	38.05	39.31	38.45	0.05	0.30

curves and their relative merits of these three schemes show similar trend on all these four test sequences, and, thus, only the *Foreman* and *News* sequences are displayed in Tables I and II. The *News* plot which has the largest variation is also displayed in Fig. 2.

As shown in Tables I and II, the TCQC scheme has the least PSNR variation as compared to JM and constant QP. It has the highest minimum PSNR and the lowest maximum PSNR. The constant QP method is the simplest conceptually but its overall PSNR is often lower; it has pretty low PSNR variation but not the lowest. Generally, the complexity of constant QP method is much lower than that of TCQC. To ensure a consistent frame-to-frame quality, TCQC has a lower PSNR than JM 7.6, but the difference is often less than 0.5 dB. As shown in Subsection IV-C, the average PSNR gets higher if the δ_P constraint is loosen. Also shown in Tables I and II are the decoder buffer delay [T_d defined in (2)], which avoids buffer underflow.

Simulation results also show that our minimum and maximum PSNR values are very close. Therefore, it is possible to narrow down the PSNR range for further complexity reduction in our algorithm. Empirically the JM average PSNR, $P_{\text{avg},\text{JM}}$, is a good estimate for the TCQC average PSNR. Extensive simulation results conclude that typically the PSNR range can be approximated by $(P_{\text{avg},\text{JM}} - 1, P_{\text{avg},\text{JM}} + 1)$.

Fig. 2 depicts the frame-to-frame PSNR plots for the *News* sequence at two different bit rates. The TCQC PSNR curve has no drop at the GOP boundaries or at scene changes. It has the smoothest shape among these three curves. The overall PSNR performance of JM 7.6 is the best but it has a large swing of more than 3 dB in PSNR across the entire sequence. One may notice that the first few frames of the TCQC algorithm have higher PSNR. This agrees with the well-known observation that a good I frame leads to better P frames in a GOP. As discussed earlier, the Viterbi search provides the optimal solution under the given assumptions and constraints. Therefore, although the average PSNR of TCQC is slightly lower than that of JM, TCQC offers the best average PSNR under the consistent quality constraint.

B. LCQC Performance Comparison With LPF and MultiStage Algorithms

In this subsection, two recent well-performed rate-control algorithms, LPF in [17] and MultiStage in [19], are simulated and compared to our LCQC algorithm. The basic idea behind LPF (low-pass filtering) is to smooth out (low-pass filtering) the distortion curve by reallocating the bits of frames inside a moving

TABLE III
COMPARISONS OF PSNR, BIT RATE, AND COMPLEXITY FOR LCQC, MULTISTAGE, AND LPF ALGORITHMS ON *NEWS* AT THREE BITRATES

Target Rate	Method	Rate(% error)	PSNR(dB)				Time(hr)
			Min	Max	Avg	Var	
24 kbps	LCQC	23.82 (-0.76%)	28.29	29.22	28.57	0.017	2.33
	MultiStage	24.11 (+0.44%)	28.19	28.6	28.36	0.002	3.90
	LPF	23.99 (-0.06%)	28.15	28.62	28.38	0.004	0.41
64 kbps	LCQC	63.93 (-0.11%)	34.35	35.09	34.70	0.018	1.63
	MultiStage	62.49 (-2.36%)	34.35	34.91	34.60	0.006	2.25
	LPF	62.66 (-2.10%)	34.22	34.88	34.54	0.014	0.41
112 kbps	LCQC	111.50 (-0.45%)	37.82	38.61	38.31	0.020	1.17
	MultiStage	107.82 (-3.73%)	37.70	38.45	38.06	0.009	1.42
	LPF	107.19 (-4.30%)	37.70	38.65	38.14	0.021	0.41

TABLE IV
COMPARISONS OF PSNR, BIT RATE, AND COMPLEXITY FOR LCQC, MULTISTAGE, AND LPF ALGORITHMS ON *TABLE TENNIS* AT THREE BIT RATES

Target Rate	Method	Rate(% error)	PSNR(dB)				Time(hr)
			Min	Max	Avg	Var	
24 kbps	LCQC	23.84 (-0.65%)	26.22	26.84	26.42	0.019	2.33
	MultiStage	25.35 (+5.61%)	26.71	27.31	26.91	0.005	5.94
	LPF	24.31 (+1.29%)	26.18	27.51	26.93	0.018	0.41
64 kbps	LCQC	63.61 (-0.61%)	32.10	33.09	32.48	0.022	2.12
	MultiStage	63.17 (-1.3%)	32.10	32.77	32.47	0.007	3.25
	LPF	62.66 (-2.10%)	32.09	32.79	32.45	0.011	0.41
112 kbps	LCQC	111.65 (-0.31%)	34.92	36.15	35.41	0.028	1.33
	MultiStage	108.76 (-2.89%)	34.92	35.50	35.25	0.007	2.32
	LPF	105.70 (-5.63%)	34.71	35.34	35.01	0.013	0.41

TABLE V
EFFECT OF QUALITY VARIATION CONSTRAINT ON PSNR AND COMPLEXITY FOR THE LCQC ALGORITHM ON THREE SEQUENCES, *FOREMAN*, *TABLE TENNIS*, AND *NEWS* AT THREE QUALITY CONSTRAINTS. $P_{\Delta} = P_U - P_L = 2$ dB

Video	Target Rate	Quality Constraint(dB)	PSNR(dB)				Time(hr)
			Min	Max	Avg	Var	
Foreman	24kbps	0.2	25.76	26.48	26.01	0.017	2.45
		0.4	25.67	26.91	26.07	0.055	2.51
		1.0	25.34	27.48	26.32	0.319	2.56
Table Tennis	64kbps	0.2	32.10	33.09	32.47	0.022	2.12
		0.4	32.09	33.15	32.50	0.049	2.23
		1.0	31.75	33.62	32.81	0.138	2.31
News	112kbps	0.2	37.82	38.61	38.31	0.020	1.17
		0.4	37.75	38.70	38.34	0.027	1.24
		1.0	37.93	39.31	38.47	0.062	1.29

time window. A quite accurate model that relates the smoothed distortion and the smoothed bit rate is proposed in [17].

The MultiStage algorithm is aiming at the constant quality target. A 2-stage iterative procedure is proposed [19]. Given a set frame bits, the *Target rate stage* encodes each frame with the given bits. Given the average PSNR of all frames, the *Constant quality stage* tries to encode every frame to reach the average PSNR by adjusting QP. If either of the following two stop conditions is satisfied, the algorithm terminates: a) the difference between the maximal and the minimal PSNR value in a sequence for the quality stage and b) the difference between coded bits and

the target bits for the rate stage. In our experiment, the threshold values are 0.5 dB and 2% for the quality stage and the rate stage, respectively. The parameters used by the LCQC algorithm are: $R_T - R(\lambda)/R_T \leq 0.02$, $\delta_P = 0.2$ dB, and $P_{\Delta} = P_U - P_L = 2$ dB.

As shown in Tables III and IV, typically, the LCQC algorithm can match the bit budget very well. The MultiStage algorithm usually has a bit rate mismatch especially at low rates, which are consistent with the report in [19]. The LPF algorithm has a bit rate mismatch too. As discussed in [17], the coding bits converge to the budget bits when the sequence length goes to in-

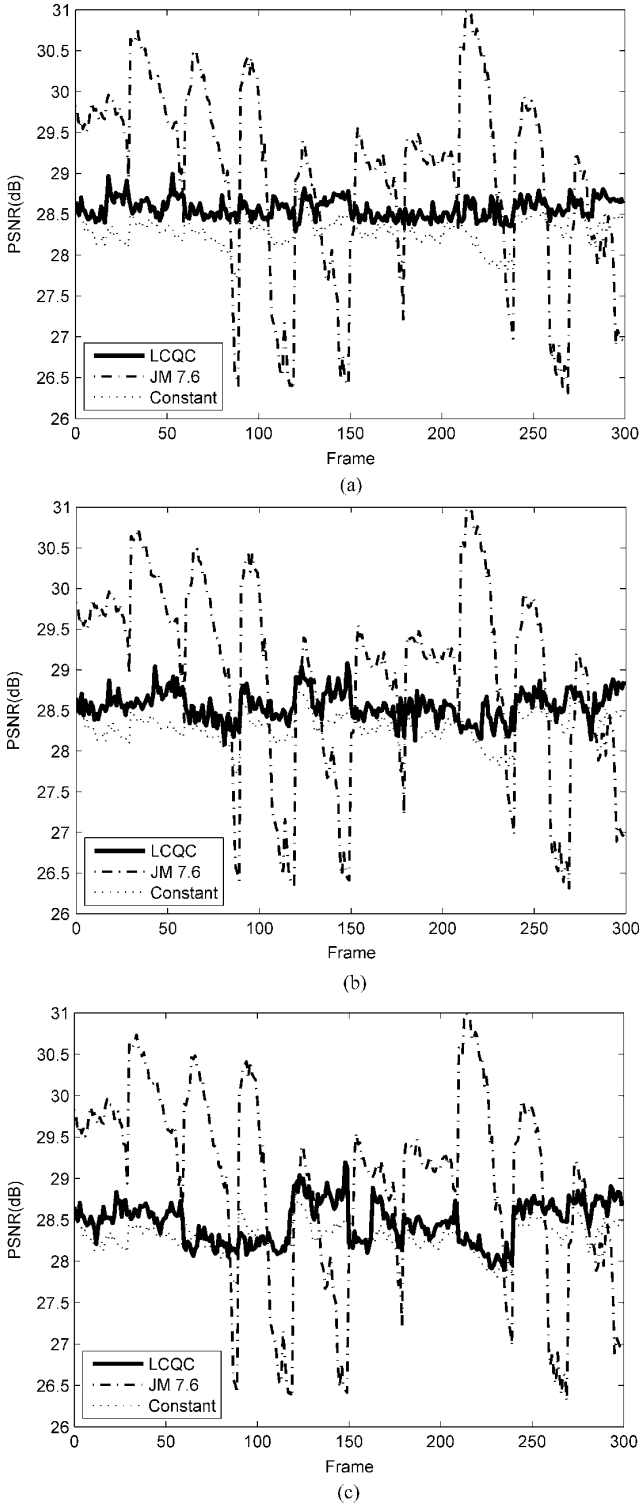


Fig. 3. LCQC results for *News* at 24 kbps and $P_{\Delta} = 2.0$ dB: (a) $\delta_P = 0.2$ dB. (b) $\delta_P = 0.4$ dB. (c) $\delta_P = 1.0$ dB.

finity. Often, the LCQC algorithm has the largest average PSNR and its PSNR variance is controlled at around 0.02 consistently at all rates because the frame quality variation is limited to a range between $-\delta_P$ and δ_P . That is, the PSNR is accurately controlled by adjusting the quality variation parameter. In contrast, the LPF and the MultiStage algorithms try to achieve the constant quality goal only. The LCQC complexity (CPU time)

TABLE VI
EFFECT OF CLUSTER SIZE P_{Δ} ON THE PSNR LOSS FOR THE TCQC ALGORITHM ON *FOREMAN* AND *TABLE TENNIS* AT THREE CLUSTER SIZES AND $\delta_P = 0.4$ dB

Video	Target Rate	Cluster Size (dB)	PSNR(dB)	
			Avg	Granularity Loss
Foreman	112 kbps	0.1	35.38	0.00
		0.2	35.29	0.09
		0.4	35.01	0.37
Table Tennis	24 kbps	0.1	26.74	0.00
		0.2	26.73	0.01
		0.4	26.62	0.12

lies in between those of MultiStage and LPF, whereas LPF has the smallest complexity. Both the LCQC and the MultiStage algorithms have a larger complexity at low bit rates due to the large number of iterations for convergence.

C. Effects of Quality Variation Constraint on PSNR and Complexity

One important feature of our schemes is the flexibility of adjusting the picture quality variation over time. Our schemes achieve the MINAVE goal in (1) when $\delta_P = \infty$. It produces the constant quality pictures when δ_P approaches 0 dB. Generally, if δ_P is smaller than 0.4 dB, a consistent quality solution is practically obtained. By adjusting the δ_P value in the range of $[0.4, \infty]$, we obtain a solution in between the constant quality and the MINAVE.

The disadvantage of using a large δ_P value is to increase the number of branch expansions and active nodes. It has a much less impact on the LCQC algorithm since LCQC does not have the trellis structure. Table V shows the test results. Indeed, its computational load increases only slightly from a small δ_P to a large δ_P .

Fig. 3 is the frame PSNR plot for the *News* sequence at different quality variation constraints. Simulation results show that a larger δ_P value leads to larger picture variation but produces a higher PSNR. As shown in Fig. 3(b), the PSNR curve produced by LCQC has little variation at the beginning of the sequence as compared to that of TCQC in Fig. 2(a). It shows that the LCQC algorithm generates even more smooth PSNR outputs.

D. Effects of Cluster Size on PSNR and Complexity

Table VI shows the TCQC results at various P_{Δ} values. As expected, the average PSNR value decreases when P_{Δ} gets larger. The *granularity loss* is defined as the absolute PSNR differences between $P_{\Delta} = 0.1$ dB case (very fine granularity) and the larger P_{Δ} cases. When the cluster size is very small ($P_{\Delta} = 0.1$ dB), we essentially achieve the best possible results without PSNR loss due to the use of cluster. As expected, the granularity loss is getting larger as the cluster size is larger than 0.1 dB.

Since LCQC does not have trellis structure, LCQC has no granularity loss. However, on the other hand, the LCQC formulation is an approximation to the integer programming problem [11]. Also, in the lambda search procedure, we stop at a given tolerance. Therefore, there is a performance loss due to the use of Lagrange cost and tolerance. Table VII shows the test results of TCQC and LCQC at the same quality variation of $\delta_P = 0.4$

TABLE VII
COMPARISONS OF AVERAGE PSNR, BIT RATE, AND COMPLEXITY IN TCQC AND LCQC ALGORITHMS FOR THREE SEQUENCES

Video	Target Rate	Rate(% error)		Avg. PSNR(dB)		Time(hr)	
		TCQC	LCQC	TCQC	LCQC	TCQC	LCQC
Foreman	24 kbps	23.83 (-0.73%)	23.69 (-1.30%)	26.11	26.07	8.15	2.51
	64 kbps	63.38 (-0.97%)	63.74 (-0.41%)	32.30	32.30	4.08	1.82
	112 kbps	111.46 (-0.48%)	111.87 (-0.11%)	35.38	35.38	3.21	1.39
News	24 kbps	23.95 (-0.20%)	23.90 (-0.42%)	28.81	28.55	7.56	2.42
	64 kbps	62.95 (-1.65%)	63.23 (-1.20%)	34.81	34.68	3.53	1.69
	112 kbps	109.71 (-4.24%)	111.57 (-0.38%)	38.41	38.34	1.84	1.24
Table Tennis	24 kbps	23.70 (-1.25%)	22.97 (-4.28%)	26.74	26.38	7.98	2.47
	64 kbps	63.17 (-1.30%)	63.68 (-0.51%)	32.61	32.50	5.26	2.33
	112 kbps	111.66 (-0.30%)	111.53 (-0.42%)	35.4	35.37	3.29	1.41

dB and the same PSNR range of $P_U - P_L = 2$ dB. The cluster size is 0.1 dB for TCQC. As expected, TCQC is slightly better but the PSNR difference is typically less than 0.3 dB. Again, LCQC is much faster in speed.

V. CONCLUSION

In this paper, we realize the triple goal of producing consistent quality videos, minimizing the total distortion and meeting the bit budget strictly. Moreover, this framework can flexibly provide a solution in between the MINAVE and constant quality extremes. Two algorithms are proposed to find the optimal and consistent quality solution. Inspired by the previous work, a trellis-based quality control scheme is firstly proposed. This approach provides a nearly optimal solution (the resulting total distortion is minimized) for a given bit rate budget on a dependent coding platform. The second algorithm is developed based on the Lagrange multipliers method. We impose the consistent quality constraint on this formulation and also we design a fast procedure to find the optimal solution. As compared to the trellis-based algorithm, it runs much faster and has a performance very close to the former. Simulation results show that both approaches have the largest PSNR average at a slight PSNR variation as compared to the other published consistent quality proposals and have a much smaller PSNR variation at a slight average PSNR loss as compared to the MPEG JM rate control. In addition, only the proposed algorithms can strictly meet the target bit budget requirement.

Due to the interframe dependent consideration, two proposed schemes have rather high computational complexity. Therefore, they are targeting at off-line applications such as Internet video streaming and DVD playback, in which the coding performance has a higher priority than the complexity. More powerful techniques that reduce the computational complexity are under development.

REFERENCES

- [1] in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Joint Final Committee Draft (JFCD) of Joint Video Specification (ITU-T Rec. H.264—ISO/IEC 14496-10 AVC), JVT-D157, 4th Meet.*, Klagenfurt, Germany, Jul. 2002.
- [2] T. Wiegand, G.-J. Sullivan, G. Bjontegarrd, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [3] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, Nov. 1998.
- [4] A.-E. Mohr, "Bit allocation in sub-linear time and the multiple-choice knapsack problem," in *Proc. IEEE Data Compression Conf.*, Mar. 2002, pp. 352–361.
- [5] 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.
- [6] H.-M. Hang and J.-J. Chen, "Source model for video transform coder and its application—Part I: Fundamental theory," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 4, pp. 287–298, Apr. 1997.
- [7] J.-J. Chen and H.-M. Hang, "Source model for video transform coder and its application—Part II: Variable frame rate coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 4, pp. 299–311, Apr. 1997.
- [8] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 246–250, Feb. 1997.
- [9] Z. He and S.-K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.
- [10] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with application to multi-resolution and MPEG video coders," *IEEE Trans. Image Process.*, vol. 3, no. 9, pp. 533–545, Sep. 1994.
- [11] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 26–40, Jan. 1994.
- [12] Y. Sermadevi and S.-S. Hemami, "Efficient bit allocation for dependent video coding," in *Proc. IEEE Data Compression Conf.*, Mar. 2004, pp. 232–241.
- [13] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 446–459, Aug. 1998.
- [14] G.-M. Schuster, G. Melnikov, and A.-K. Katsaggelos, "A overview of the minimum maximum criterion for optimal bit allocation among dependent quantizers," *IEEE Trans. Multimedia*, vol. 1, no. 3, pp. 3–17, Mar. 1999.
- [15] Y. Yu, J. Zhou, Y. Wang, and C.-W. Chen, "A novel two-pass VBR coding algorithm for fixed-storage application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 345–356, Mar. 2001.
- [16] Y. Sermadevi and S.-S. Hemami, "Lexicographic bit allocation for MPEG video coding," in *Proc. IEEE Data Compression Conf.*, Mar. 1997, pp. 101–110.
- [17] Z. He, W. Zeng, and C.-W. Chen, "Low-Pass filtering of rate-distortion functions for quality smoothing in real-time video communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 973–981, Aug. 2005.
- [18] B. Xie and W. Zeng, "A sequence-based rate control framework for consistent quality real-time video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 1, pp. 56–71, Jan. 2006.

- [19] N. Cherniavsky *et al.*, "MultiStage: a MINMAX bit allocation algorithm for video coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 59–67, Jan. 2007.
- [20] S.-Y. Lee and A. Ortega, "Optimal rate control for video transmission over VBR channels based on a hybrid MMAX/MMSE criterion," in *Proc. IEEE Int. Conf. Multimedia Expo*, Aug. 2002, vol. 2, pp. 93–96.
- [21] J.-J. Chen and D. W. Lin, "Optimal bit allocation for coding of video signals over ATM networks," *IEEE J. Sel. Areas Commun.*, vol. 3, no. 8, pp. 1002–1015, Aug. 1997.
- [22] D.-K. Kwon, M.-Y. Shen, and C.-C. Jay Kuo, "Rate control for H.264 video with enhanced rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 5, pp. 517–529, May 2007.
- [23] W.-Y. Lee and J.-B. Ra, "Fast algorithm for optimal bit allocation in a rate-distortion sense," *Electron. Lett.*, vol. 32, no. 20, pp. 1871–1873, Sep. 1996.



Kao-Lung Huang received the B.S. and M.S. degrees from Chung Cheng Institute of Technology, Taoyuan, Taiwan, R.O.C., in 1985 and 1989, respectively, both in electrical engineering. He is currently pursuing the Ph.D. degree in electrical engineering at the National Chiao Tung University, Hsinchu, Taiwan.

Since 1989, he has been with the Chung Shan Institute of Science and Technology (CSIST) as a Member of Technical Staff. His research interests include video coding, image/radar signal processing algorithms, and multimedia communication systems.



Hsueh-Ming Hang (S'79–M'84–SM'91–F'02) received the B.S. and M.S. degrees from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984.

From 1984 to 1991, he was with AT&T Bell Laboratories, Holmdel, NJ, and then he joined the Electronics Engineering Department, NCTU, in December 1991. He took a leave from NCTU and has been the Dean of the Electrical Engineering and

Computer Science College, National Taipei University of Technology (NTUT), since August 2006. He is a co-editor and contributor of the *Handbook of Visual Communications* (Academic). He holds 11 patents (R.O.C., U.S., and Japan) and has published over 150 technical papers related to image compression, signal processing, and video codec architecture. His research interests include multimedia compression, image/signal processing algorithms and architectures, and multimedia communication systems.

Dr. Hang was an associate editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING (1992–1994), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (1997–1999), and is currently an associate editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING again. He is a recipient of the IEEE Third Millennium Medal, a Fellow of IET, and a member of Sigma Xi.