Journal of VLSI Signal Processing 20, 233–250 (1998)

The Impact of Rate Control Algorithms on System-Level VLSI Design

SHEU-CHIH CHENG AND HSUEH-MING HANG
Department of Electronics Engineering, National Chiao Tung University, 1001 Ta-Hsueh Rd, Hsinchu 300, Taiwan, Republic of China

Received October 17, 1997; Revised May 13, 1998

Abstract. This paper presents an evaluation of rate control algorithms from a system-level VLSI design viewpoint. Rate control in video coding has a significant influence on the coded bit rate and image quality. Many rate control algorithms have been proposed mainly focusing on the optimal rate-distortion performance without considering their performance on the VLSI implementation. The purpose of this study is not to propose a hardware architecture for any specific algorithm but to study the algorithm impact on hardware design. Based on our finding, a system designer should choose an algorithm not only good in rate control performance but also good in hardware implementation. When implementing and comparing a few rate control algorithms using a generic processor structure, we found that, in addition to the ordinary computational complexity, the internal buffer size is also very critical in VLSI realization. Several picture sequences have been tested including one sequence constructed specifically to simulate a difficult case for rate control. In this paper, three different types of popular rate control algorithms have been analyzed based on their picture quality, the internal buffer size, and the hardware cost. The methodology and results presented here provide useful guidelines for selecting an appropriate rate control algorithm for system-level VLSI designers.

1. Introduction

There are tradeoffs among various hardware cost and performance factors in designing a VLSI chip [1]. Since chip design and layout process are time-consuming and costly, it is very desirable to be able to predict the overall system performance of a high-level algorithm before the circuit layout is fully deployed. The purpose of this paper aims at studying the impact of various types of rate control algorithms on VLSI design. For digital video transmitted over a bandlimited channel, such as advanced digital TV, CD-ROM, and digital video disk recording, rate control is one of the critical elements to determine the picture quality and compression efficiency in a video coding system. In addition, in MPEG1/2 coding the rate control also plays the role of preventing the output data buffer from overflow or underflow. In realization, the main function of rate control is to distribute the assigned bits properly among image macroblocks through the adjustment of quantization stepsize (*mquant*). One important element in the above process is designing a good picture complexity measuring function so that the best *mquant* can be chosen to produce a good picture quality and to meet the channel requirement.

The rate control problem is particularly important for constant bit rate (CBR) transmission. To meet the constant bit-rate requirement, the most straightforward rate control algorithm is the buffer-feedback control algorithm, in which the *mquant* is mainly decided by the buffer fullness. However, the *mquant* determined merely by the status of buffer is not optimal in the rate-distortion (R-D) sense and this process may require a large size of internal (on-chip) buffer to temporarily store the compressed bitstream. To achieve the best objective quality, many rate control algorithms have been devised to optimize the rate-distortion performance without considering the cost of high computational complexity and/or large-size internal buffer. From the system-level design viewpoint, it is essential to consider the impact of the rate control algorithm on hardware implementation. Therefore, the purpose of

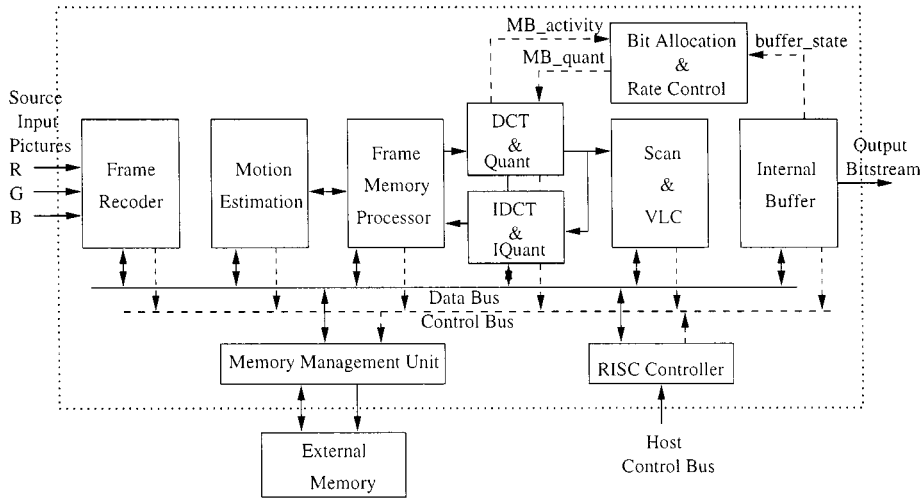


Figure 1. Block diagram of an MPEG encoder chip.

this study is to compare various types of rate control algorithm from the VLSI implementation viewpoint. In order to achieve an objective comparison, a common platform is constructed for evaluation purpose. Based on our survey, the generic processing structure proposed in this paper is adequate in implementing many different types of rate control algorithms to interests.

The block diagram of our MPEG encoder structure is shown in Fig. 1. This structure is similar to the MPEG2 Test Model 5 (TM5) [2]. A brief description of this system is given below. Details of the system architectures are reported in [3]. The input pictures are first stored in the external memory under the control of the frame recoder unit. The memory management unit (MMU) acts as an interface between the processing units and the external RAM, because the memory access bandwidth is a bottleneck in the encoder chip design. The external RAM is an off-chip memory. A portion of it is used as the off-chip output buffer (the VBV-buffer in the standard) and the other portion is the image data memory. The motion estimation unit estimates the motion vectors used in interframe coding and the frame memory processor module (FMP) is used to perform the function of motion compensation at the encoder. It imitates the job of reconstructing the reference frames at the decoder. The RISC controller performs the functions of rate control, timing control, and other coding parameters adjustment.

Discrete cosine transform (DCT) and variable length coding (VLC) have been widely recognized as efficient means for compressing images and are adopted by many image standards. The quantization module (Quant) is essentially a division operation. The DCT coefficients are divided by a *mquan* value times a visually weighted quantization matrix. The operations of DCT, VLC and Quant are defined explicitly in the MPEG-2 standard. Hence, algorithm variations will by and large appear in the Motion Estimation and the Rate Control units. A study of the impact of motion estimation algorithms on VLSI design has been reported in [4]. We will concentrate on the rate control algorithms in this paper. Often neglected by the algorithm designers is the internal (on-chip) output buffer that temporarily stores the compressed bitstream. Potentially, the MPEG compressed bits produced by the VLC unit can be greater than several thousands bits per macroblock, a burst of several hundreds Mbps in bandwidth. (In the meanwhile, many other encoder units such as the Motion Estimator have to access the external data at large quantity too.) This poses impractical requirements on both I/O bandwidth and external RAM. Hence, an internal output buffer is introduced to smooth out the data transferring rate between the VLC unit and the external RAM. In addition, the rate control unit is one of the dominant units in chip area. Based on our study [3], the silicon area of the rate control unit is comparable to that of the DCT or FMP units.

The rest of this paper is organized as follows. Section 2 describes the rate control algorithms examined. Section 3 contains the main theme of this paper, which discusses the computational complexity of the evaluated rate control algorithms and their silicon area.

The picture quality comparison of various rate control algorithms is presented in Section 4. Section 5 briefly summarizes our work.

2. Rate Control Algorithms

Rate control is an essential element in a video coding system in order to transmit the coded bitstream over a constant bit-rate channel. The goal of rate control algorithm is to efficiently distribute the coded bits properly to each coded image block at a given total bits budget, so that the channel and the decoder buffer requirements are satisfied. In MPEG1/2 image encoding, the quantization scale (*mquant*) is one of the key parameters that determines the picture quality and the coding bits. A small quantization scale offers a lower distortion in image picture, but produces a larger amount of coding bits. On the other hand, a large quantization scale though generates fewer bits, it may produce serious distortion. Thus, the basic operation of a rate control algorithm is to choose a proper *mquant* to meet the bits budget requirement. In general, the rate control algorithm consists of two operations, namely, the bit allocation and the quantizer selection. The bit allocation unit estimates the number of bits available for coding the next picture and distributes the available bits to picture blocks. The quantizer selection measures the image contents of an image macroblock and decides a proper *mquant*.

In the following analysis, we assume that an input image sequence has N_{pic} picture frames and each frame is partitioned into N_{mb} macroblocks. Suppose the target bit counts for coding the kth frame is R_k. Let R denote the constant output bit-rate; then, the relation between the given bits quota and frame bits distribution is:

$$\sum_{k=1}^{N_{\text{pic}}} R_k = N_{\text{pic}} \cdot \frac{R}{f_r}, \tag{1}$$

where f_r is the frame rate. Different approaches have been taken to determine R_k, such as image complexity measures, bits estimation model, and optimal bit allocation methods. In TM5, the R_k value is determined by a DCT complexity measure and the quantization scale of the most recent picture of the same type. Puri and Aravind [5] have proposed a bits modeling approach to decide R_k under an eight-scene complexity classifier, where the parameters of bits model are empirically determined and pre-stored in the codec. Most compression standards (such as MPEG1/2) allow using different quantization stepsizes for the DCT coefficients in different portions of a picture. Thus, the quantizer selection unit is needed to determine the quantization stepsize for each macroblock to meet the given bits budget.

Let r_i be the coded bits of the ith macroblock. In general, the macroblock quantization stepsize is chosen based on the selected distortion d_i, image macroblock activity a_i, and the buffer fullness b_i. In terms of mathematical notations, the candidate *mquant* (Q_i) of macroblock i is expressed by:

$$Q_i = F(d_i, a_i, b_i) \quad \text{subject to} \sum_{j=1}^{N_{\text{mb}}} r_j(Q_j) \le R_k. \tag{2}$$

For the ease of theoretical analysis, the distortion d_i is often specified as the mean square error of the difference between the original and the coded pel. The block activity a_i is a measure of the image block content complexity such as the block energy, and the buffer fullness b_i is the number of bits in the VBV-buffer. For convenience, we define two terms for measuring the macroblock activity: (a) MBV (minimum block variance), which is the minimum variance among the four luminance DCT blocks in a macroblock, and (b) SCM (sum of DCT coefficients in a macroblock), which is the sum of all the DCT coefficients inside a macroblock without DC coefficient. That is,

$$\begin{aligned} MBV &= 1 + \min_{j=1,\ldots,4} \left\{ \sum_{k=1}^{n} (c_{j,k} - avg\_c_j)^2 \right\}, \\ SCM &= \sum_{j=1}^{6} \sum_{k=2}^{n} |c_{j,k}|, \end{aligned} \tag{3}$$

where $\{c_{j,k}, k = 1, \ldots, n\}$ is the DCT coefficients of the jth block, n is the number of coefficients in a block, and avg_c_j is the block average DCT coefficient value. In general, we need $4n$ subtractions, $4n$ additions, and $4n$ multiple operations to compute an MBV. For SCM, it requires $6n$ additions and $6n$ absolute value operations.

In a typical rate control algorithm, the *mquant* value is selected based on $F(d, a, b)$. To our knowledge, the existing rate control algorithms can be classified into three groups according to their *mquant* determination strategy: (1) buffer-feedback method, (2) budget planning method based on simple bits models, and (3) optimal bit allocation method in rate-distortion sense. In

fact, the so-called optimal bit allocation algorithm can be viewed as a special case of the budget planning approach with a computational-intensive bits model to produce the minimum rate-distortion. The following subsections describe the operations of these rate control algorithms. However, a complete rate control scheme may contain elements from more than one strategy. For example, in the MPEG coding structure, there are three levels of bits budgets: (a) group of pictures (GOP), (b) pictures and (c) macroblock. Different or mixed strategies may be used at different levels.

2.1. Buffer-Feedback Method

To meet the exact target bits budget, the most straightforward rate control scheme is the buffer-feedback method. In principle, a pure buffer-feedback control does not assess the image block content. The current block quantization step is decided only by the current buffer fullness. However, it would produce a more uniform quality picture PSNR if it also takes the image block content into account. An example of buffer-feedback rate control algorithm is the well-known Reference Model 8 (RM8) of H.261. The current block quantization step is a linear function of the buffer fullness [6]. Another example but with elements of budget planning approach is the MPEG2 Test Model 5 (TM5). In addition to buffer fullness measure, it also contains an image content measuring mechanism. The TM5 algorithm basically has three steps: (1) allocate a frame target bit budget according to the image complexity of the most recent picture of the same type, (2) select a nominal slice quantization scale according to the buffer fullness, and (3) adjust the quantization scale of each macroblock by examining to the spatial activity of its luminance blocks.

In TM5, the target bits for the next picture in a GOP is computed by

$$R_t = \max\left\{\frac{R_{\text{GOP}} \cdot \frac{X_t}{K_t}}{\sum_{i\in\{I,P,B\}} N_i \cdot \frac{X_i}{K_i}}, \frac{R}{8\cdot f_r}\right\}, \quad (4)$$

where X_i and N_i are the measured complexity of the most recent picture and the number of remaining frames, respectively, of the same i type; R_{GOP} is the remaining number of bits assigned to the current GOP. K_t takes value from (K_I, K_P, K_B), where K_I, K_P, and K_B are the "universal" constants depending on the quantization matrix; their values in TM5 are assigned to be 1, 1, and 1.4, respectively. The *mquant* is mainly determined by the current buffer fullness (b_i), and then it is modified by the "normalized MBV" ($N\_act(MBV)$) of an macroblock. In this paper, the last two steps are combined in order to characterize the quantizer selection through the macroblock complexity function. The quantization scale (Q_i) for the ith block is thus calculated by:

$$Q_i = F_{\text{TM5}}(a_i, b_i) = \frac{31\cdot b_i}{r}\cdot\{N\_act(MBV_i)\}, \quad (5)$$

where r is the "reaction parameter" decided experimentally and $N\_act(MBV_i)$ is the "normalized MBV" for the ith macroblock,

$$N\_act(MBV_i) = \frac{2\cdot MBV(a_i) + avg\_act}{MBV(a_i) + 2\cdot avg\_act}, \quad (6)$$

where *avg_act* is the average value of $MBV(a_i)$ of the last coded picture.

2.2. Budget Planning Rate Control Algorithm

The budget planning rate control algorithm is another popular method to solve the buffer control problem. The original concept is a two-pass algorithm. Every image block complexity is measured in the first pass. Then a proper bits budget (and *mquant*) is assigned to each block according to their complexity and other factors such as visual effect. This method is developed based on the assumption that the coded bit counts can be predicted from the image content complexity. Therefore, a pure budget planning scheme examines the buffer fullness only once per frame or per GOP and then it is an open-loop control. For example, a bits model can be constructed for predicting the coded bits for each picture or for each macroblock based on their DCT coefficients and the selected *mquant* values [5, 7]. However, it may produce more or fewer bits if the bits model is not accurate enough to predict the actual coded bits. In addition to using a good bits model, it also adjusts the macroblock bits budget to meet the frame target bits R_t. Several bits models have been suggested in the literature [8, 9]. These bits models are typically constructed through various data fitting techniques using an off-line training process. These bits models are determined by experiments and pre-stored at the encoder. For practical purpose, a one-pass version is often used. In this usage, the bits model produces an estimate of the target bit counts for the next uncoded pictures. In order to reduce the computational

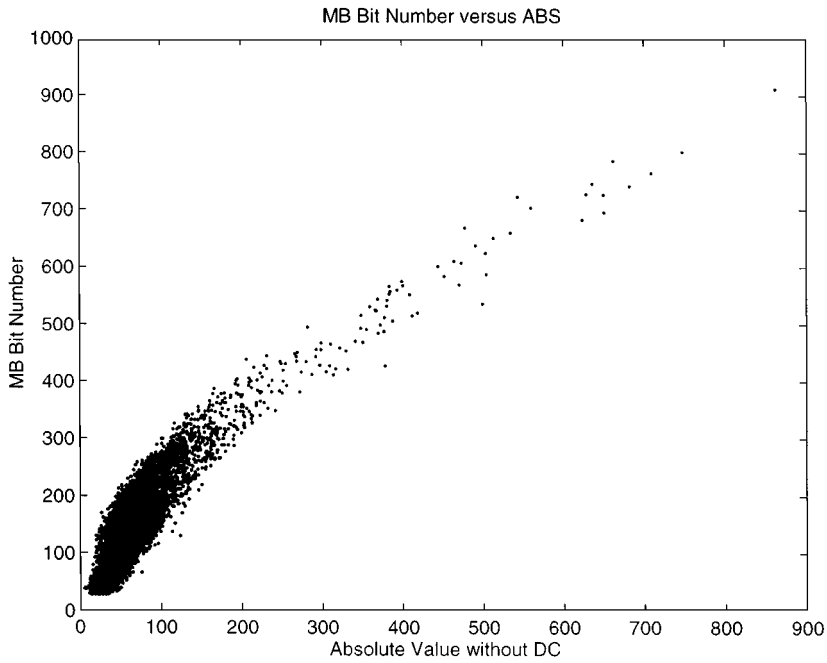


Figure 2. Macroblock bits versus the sum of absolute DCT coefficients without DC terms for the *Football* sequence.

complexity, a linear bits model is often chosen and it provides a reasonably good result. The "total sum of the absolute value of DCT coefficients without DCT term" [8] was found to be a rather accurate block complexity measure. Figure 2 shows the experimental result of the coded bits and its associated DCT activity (block complexity) in the TM5 algorithm for intra-coded (I) pictures. It can be observed that a proper piecewise linear bits model can fit the actual coded bits rather well. This piecewise linear bits model is expressed by

$$N_{\text{bit}} = c_1 \cdot \frac{SCM(a_i)}{Q_i} + c_0, \tag{7}$$

where c_1 and c_0 are the model parameters that may be adjusted adaptively. The piecewise linearity relation is also valid for the predictive-coded (P) and bidirectionally predictive-coded (B) pictures with the pre-trained parameters shown in Table 1.

In a budget planning algorithm, we also need to allocate target bits for picture frames. In order to be consistent with the macroblock quantizer selection, a frame bit allocation strategy different from that of TM5 is proposed for the budget planning algorithm. A second piecewise linear bits model is used to predict frame bits and its coefficients are listed in Table 1. On the other hand, to compare the coding performance, we can also take the frame bits budget estimated by TM5 followed by the budget planning method applied only to the macroblocks. The PSNR performances of budget planning algorithms using two types of frame bit allocation strategies are shown in Fig. 3. The results indicate that the frame bits allocation in the TM5 algorithm is less adequate for the budget method.

Let us summarize the budget planning rate control algorithm procedure below.

Table 1. Pre-trained parameters for budget planning algorithm (channel rate = 5 Mbps, CCIR-601 picture size).

Layer	Type	Activity	C_1	C_0
Macroblock	I	≤ 100	2.2102	19.6105
		>100	1.5185	64.5496
	P	≤ 100	2.2437	−21.1716
		>100	2.1418	−25.7073
	B	≤ 100	2.3697	−26.434
		>100	1.9619	2.36
Picture	I	≥ 0	140	179030
	P	≥ 0	130	161660
	B	≥ 0	320	117710

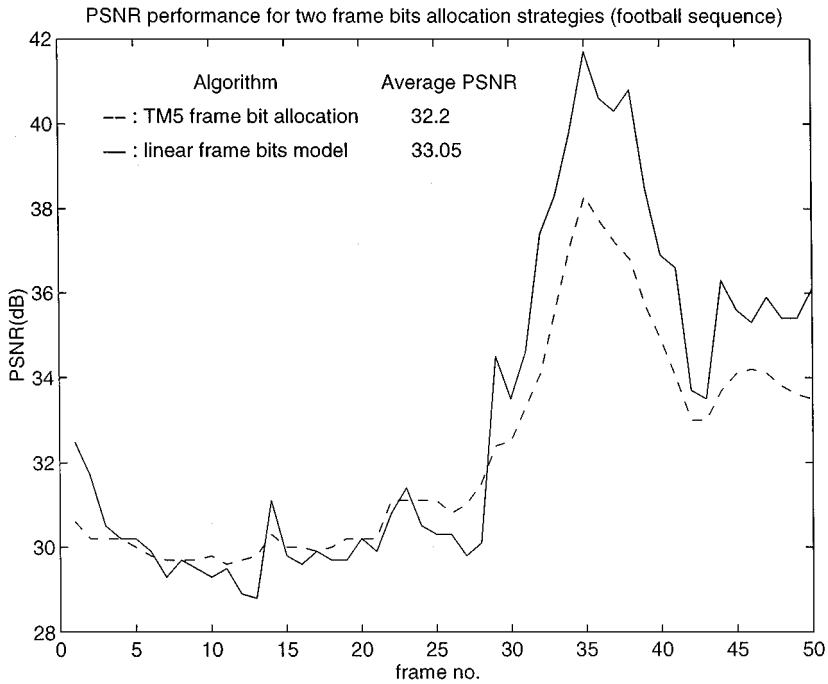


Figure 3. PSNR performance comparison for two frame bit allocation strategies in the budget planning scheme.

1. Picture (frame) level bit allocation: Allocate the target frame bits (R_t) according to the image content of the most recent picture of the same type,

$$R_t = c_1 \cdot \frac{pic\_act}{pic\_qan} + c_0, \tag{8}$$

where *pic_act* and *pic_qan* are, respectively, the sums of macroblock DCT activities and quantization stepsizes in a picture frame.

2. Macroblock level quantizer selection: There are two steps in selecting Q_i,

   2.1. Allocate the macroblock bits budget based on its DCT activity:

$$r_i = r_i' \cdot \frac{\log(SCM(a_i))}{\frac{1}{N_{\text{mb}}} \sum_{k=1}^{N_{\text{mb}}} \log(SCM(a_k))}, \tag{9}$$

   where r_i' is the average number of bits allocated to the macroblock i,

$$r_i' = \frac{R_{\text{pic}}}{N_{\text{mb}} - (i-1)},$$

   in which R_{pic} denote the remaining bits assigned to the current frame and its values is initially assigned to be R_t before encoding the first macroblock in each frame. The *log* operation is inserted so that the coded image has a more uniform visual quality [9].

   2.2. Calculate the *mquant* for each macroblock,

$$Q_i = F_{\text{budget}}(a_i) = \frac{c_1 \cdot SCM(a_i)}{r_i - c_0}. \tag{10}$$

After encoding a macroblock, R_{pic} is updated by $R_{\text{pic}} = R_{\text{pic}} - r_{\text{act}}$, where r_{act} is the actual coded bits. When the bits model is accurate, the actual coded bit count is close to the predicted one. However, the model parameters are picture-dependent. We need to adjust the bits model to match the actual coded bits. The model parameters are updated by the LMS algorithm once per macroblock and/or per frame. The updating formula is

$$\begin{aligned} c_1' &= c_1 + \mu \cdot (r_{\text{act}} - r_i) \cdot \frac{SCM(a_i)}{Q_i} \\ c_0' &= c_0 + \mu \cdot (r_{\text{act}} - r_i), \end{aligned} \tag{11}$$

where μ is an updating parameter and its value is 10^{-6} at the macroblock level and 10^{-5} at the frame level.

2.3. Optimal Bit Allocation in Rate-Distortion Sense

The purpose of the optimal bit allocation algorithm is to achieve the minimum coding distortion under the given bits budget constraint. Because the quantizer and VLC operations are nonlinear and the picture content is time-varying, it is very difficult to build an accurate bits model to predict the actual coded bits. The basic idea is to evaluate the output SNR for all possible combinations of *mquant* values for one or several pictures and choose the best one. For example, if each macroblock can use 10 different stepsizes, all possible combinations of a picture frame with 300 macroblock is 10^{300}. It can be viewed as a specific budget planning algorithm without a simple explicit closed-form bits model. Many algorithms are proposed to reduce computation. For example, an integer programming approach is suggested to find the optimal rate-distortion (R-D) solution [10, 11]. The main problem of bit allocation lies in allocating bits optimally among the DCT-coded macroblocks inside a picture and among frames in a video sequence. This becomes an even more difficult problem because many popular compression standards use dependent coding, i.e., the set of available R-D operating points depends on the particular choice of quantization stepsizes in the previous macroblocks or previous frames. To solve the dependent coding problem, several algorithms use the multilevel dependency tree (or trellis) where the number of "status" nodes in the tree structure decides the quantization choices for the next to-be-coded blocks and/or frames [12]. Thus, these methods require huge buffers to store candidate R-D operation points until an independent coding unit (a complete picture for macroblock bits allocation or a GOP for frame bits allocation) is reached and all the quantization stepsizes in the tree are decided. The goal of this paper is to evaluate the performance of rate control algorithms from the VLSI design viewpoint. To simplify calculation, we compare the PSNR performance of these algorithms under the same frame bits; the frame target bits allocation in the following experiments is the same as that in the TM5 algorithm.

The modified optimal rate control algorithm is summarized as follows.

1. Compute the distortion (D_i) and the coded bits (r_i) for various quantization stepsizes (Q_i) of each macroblock (index i).
2. Calculate the Lagrange cost using the Lagrange multiplier (λ),

$$\min_{Q_i} \left[\sum_{i=1}^{N_{\text{mb}}} D_i(Q_i) + \lambda \sum_{i=1}^{N_{\text{mb}}} r_i(Q_i) \right]. \tag{12}$$

3. For a given target number of bits R_t, update λ using [13],

$$\lambda = \lambda_l \left(\frac{\lambda_l}{\lambda_u} \right)^{\left(\frac{R_l - R_t}{R_l - R_u} \right)}, \tag{13}$$

   where the R_l and R_u are the frame coded bits at the lower bound and upper bound of the Lagrange multipliers (λ_l and λ_u), respectively. The initial values of λ_l and λ_u are chosen to meet the $R_u \leq R_t \leq R_l$ condition.
4. Go to Step 2 until $\sum_{i=1}^{N_{\text{mb}}} r_i(Q_i) = R_t \pm R_{\text{tol}}$, where R_{tol} is the maximum tolerance of bits error and its value is 30 bits in this paper.

3. Complexity Analysis and Chip Area

3.1. Complexity Analysis

Two important factors are taken into account in choosing the rate control algorithms in VLSI implementation. They are (1) silicon area and (2) picture quality. These two factors are discussed in this section and in Section 4. The silicon area of a rate control algorithm can be approximated by

$$A_{\text{total}} = A_{\text{op}} + A_{\text{ibuf}} + A_{\text{ext}}, \tag{14}$$

where A_{op} is the area used for the processing unit, A_{ibuf} is for the on-chip output buffer, and A_{ext} is for the additional hardware requirement. Because of the coding delay and the massive I/O bandwidth required at the encoder, an internal output buffer (ibuf) is necessary to reduce the massive short-term peak output data rate. In hardware implementation, some rate control algorithms need additional hardware circuits; for example, the optimal rate control algorithm requires additional quantization units, VLC units, and other circuits to calculate the coded bits and distortion.

In order to have a fair comparison of different rate control algorithms, a common platform is constructed for evaluation purpose. A generic processing structure

allows a higher degree of flexibility and it is adequate for an efficient implementation of many different rate control algorithms. The silicon area of the computation unit (A_{op}) is estimated based on the statistical results of the flexible programmable architectures (100 mm^2/GOP) for video codec [14]. Thus, the silicon area A_{op} depends on the computational complexity of a rate control algorithm.

In a rate control algorithm, the required number of arithmetic operations is mainly determined by the quantizer selection operation, as listed in Table 2. In this table, the equations used in computation are listed below the column heading *Operator type* and the *Processing rate* is the corresponding processing speed required to calculate the quantizer selection. In this study, the processing elements are decomposed into four groups, namely, *add*, *mul*, *div*, and *log*, based on our flexible programmable architecture. Specially, the computational complexity of the optimal bit allocation algorithm depends on the λ calculation. The parameter n_λ denotes the number of iterations for the optimal rate control algorithm to find the best value of λ. In theory, we need to consider the worst situation in VLSI design. Without any constraint, the range of λ can be very large ranging from 0 to ∞. Thus, we adopt a version with an upper bound λ_u and a lower bound λ_l. The range between λ_u and λ_l is assumed to be 100 and we neglect the calculations needed to find a correct λ range. The value of n_λ is obtained by averaging the number of iterations in finding the target λ.

Table 2. Implementation complexity for processing unit.

Operator type			Processing rate (N_{op}/s)
Algorithm (1): Test Model 5			
Eqs. (5) and (6)	Q_i	mul	$2 \cdot N_{\text{mb}} \cdot f_r$
	MBV	add	$2 \cdot 4 \cdot n \cdot N_{\text{mb}} \cdot f_r$
		mul	$4 \cdot n \cdot N_{\text{mb}} \cdot f_r$
	N_act	add	$2 \cdot N_{\text{mb}} \cdot f_r$
	(MBV_i)	div	$N_{\text{mb}} \cdot f_r$
Algorithm (2): Budget planning algorithm			
Eqs. (9), (10), and (11)	SCM	add	$2 \cdot 6 \cdot n \cdot N_{\text{mb}} \cdot f_r$
	r_i	add	$2 \cdot N_{\text{mb}} \cdot f_r$
		mul	$N_{\text{mb}} \cdot f_r$
		div	$2 \cdot N_{\text{mb}} \cdot f_r$
		log	$N_{\text{mb}} \cdot f_r$
	Q_i	add	$N_{\text{mb}} \cdot f_r$
		mul	$N_{\text{mb}} \cdot f_r$
		div	$N_{\text{mb}} \cdot f_r$
	Adaptive	add	$6 \cdot N_{\text{mb}} \cdot f_r$
		mul	$N_{\text{mb}} \cdot f_r$
		div	$N_{\text{mb}} \cdot f_r$
Algorithm (3): Optimal bit allocation			
Eqs. (12) and (13)	Q_i	add	$2 \cdot 32 \cdot n_\lambda \cdot N_{\text{mb}} \cdot f_r$
		mul	$32 \cdot n_\lambda \cdot N_{\text{mb}} \cdot f_r$
	λ	add	$2 \cdot f_r$
		mul	f_r
		div	$2 \cdot f_r$
		log	f_r

3.2. *Internal Output Buffer*

The huge compressed bits generated by the VLC unit must be transferred to the external RAM. An internal output buffer is necessary to smooth out the irregular data rate generated by the VLC unit. The size of the internal output buffer is chosen by considering the worst case of bandwidth requirement in the memory management unit (MMU). In the MPEG encoder, five units can simultaneously issue requests to the MMU for accessing the memory bus. They are the frame recorder, the motion estimator, the DCT unit, the frame memory processor, and the VLC unit. The frame recorder unit stores the input image data (N_{FR}) in the external memory. The reference picture data and the current picture data are loaded into the motion estimator (N_{ME}) for motion vector estimation. Typically, the motion estimator only requires the luminance data. The reconstructed reference data which are the output of the frame memory processing unit is stored in the external memory (N_{FMP}). Finally, the input data of DCT (N_{DCT}) and the output data of VLC (N_{VLC}) units are also stored in the external memory. Assuming that D_{mb} and D_{ref} are the numbers of the input image data and the reference data in bytes of a macroblock, then the rates of the five required data (bytes per macroblock) are: $N_{\text{FR}} = N_{\text{FMP}} = N_{\text{DCT}} = D_{\text{mb}}$, $N_{\text{ME}} = D_{\text{mb}} \cdot \frac{4}{N_{\text{mb/blk}}} + D_{\text{ref}}$, and $N_{\text{VLC}} = D_{\text{ibuf}}$, where the D_{ibuf} is the maximum output data rate of the internal output buffer. The parameter D_{ref}, which relies on the chosen ME algorithm and internal frame buffer type as described in [4], is the number of reference data required for the ME unit. The timing requirement in the memory management block is the sum of all the above five requests. For simplicity, the average macroblock processing time duration (t_{mb}) is chosen as the time unit for the entire chip design,

and the external memory bus is assumed to be W bits. Then, the access time required for the external memory is

$$t_{\text{mem}} = t_{\text{mb}} \cdot \frac{W}{N_{\text{FR}} + N_{\text{ME}} + N_{\text{FMP}} + N_{\text{DCT}} + D_{\text{ibuf}}}. \quad (15)$$

On the other hand, the size of the internal output buffer is mainly determined by D_{ibuf}, because the data flow rate is nearly constant for all the other four units. The allowable output data rate of the internal output buffer is approximately 500 bits/macroblock, under the assumptions that the external memory access time is 50 ns, the bus width (W) is 60, and the three-step search of search range 15 is used with the type B buffer [4]. The output data rate would be somewhat smaller if a larger search range in ME is used.

To test on difficult pictures, we synthesized a so-called *Gaussian* picture sequence. The CIF-size salesman picture sequence is placed on the center of a CCIR-601 frame and is surrounded by white Gaussian random noises with variance 500, as shown in Fig. 4. Two other test sequences are CCIR-601 size *Flowergarden* and *Football*. These sequences are compressed to 5 Mbps using MPEG2 encoding program (TM5) but with different rate control algorithms. The simulation results of the on-chip buffer fullness (measured at per macroblock interval) are shown in Figs. 5–7 and the average values are listed in Table 3. These values are obtained under the assumption that the compressed data rate transferred from the internal buffer to the external RAM is 500 bits/macroblock (about 20 Mbps). In this table, *Max* is the peak value and Ave_{50} is the average of the 50 largest values. It is observed that the buffer size of the budget planning rate control algorithm is about 1/4 to 1/18 of that of TM5, and the optimal bit allocation algorithm (OB) has a much larger buffer than the other two algorithms.

Table 3. On-chip buffer size for various CCIR picture sequences.

Algorithm: / Item: / Picture sequence	TM5		BP		OB	
	Max	Ave_{50}	Max	Ave_{50}	Max	Ave_{50}
Football	2469	1762	349	310	3621	2870
Flowergarden	4679	4253	416	385	58863	57718
Test picture	6839	6370	326	295	22255	20409
Internal buffer	6370		385		57718	

3.3. Chip Area Estimation

As described earlier, the optimal bit allocation algorithm requires additional external quantization and VLC units to calculate the coded bits and additional

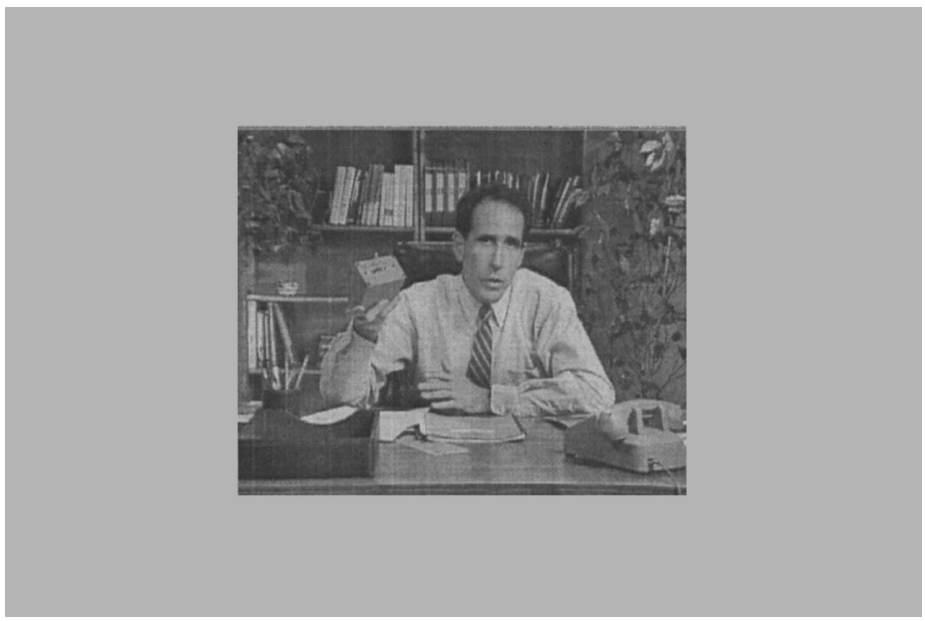


Figure 4. The original Gaussian test sequence (10th frame).

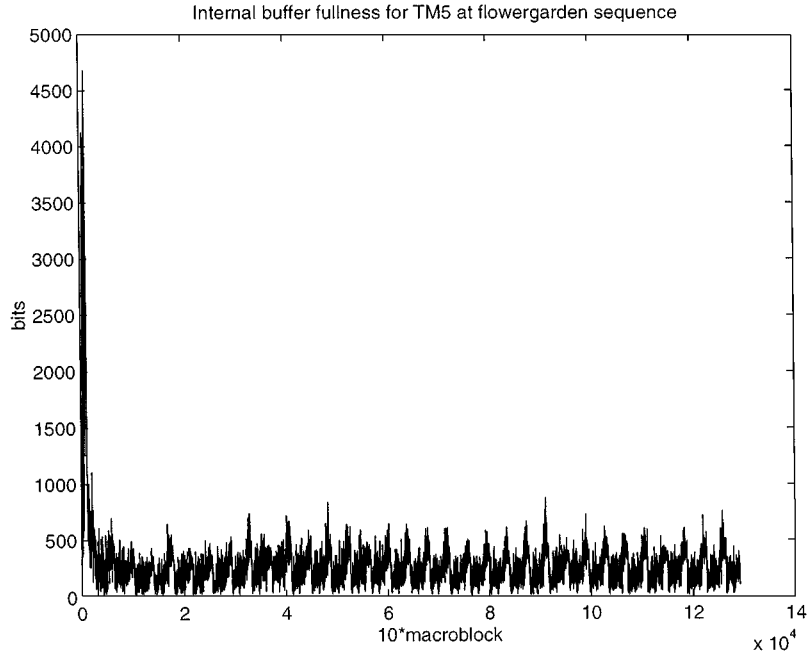


Figure 5. Internal buffer fullness of the TM5 rate control algorithm for the *Flowergarden* picture sequence.

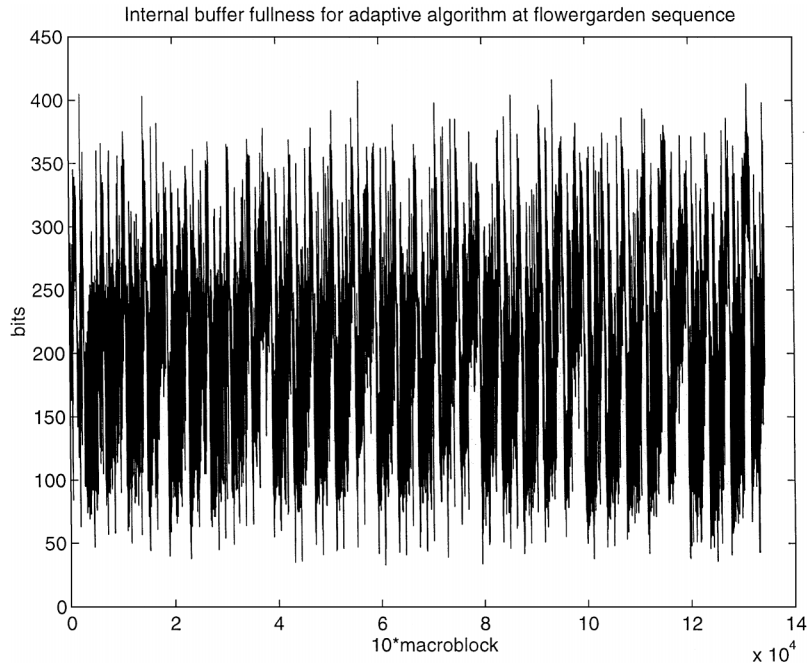


Figure 6. Internal buffer fullness of the adaptive budget planning rate control algorithm for the *Flowergarden* picture sequence.

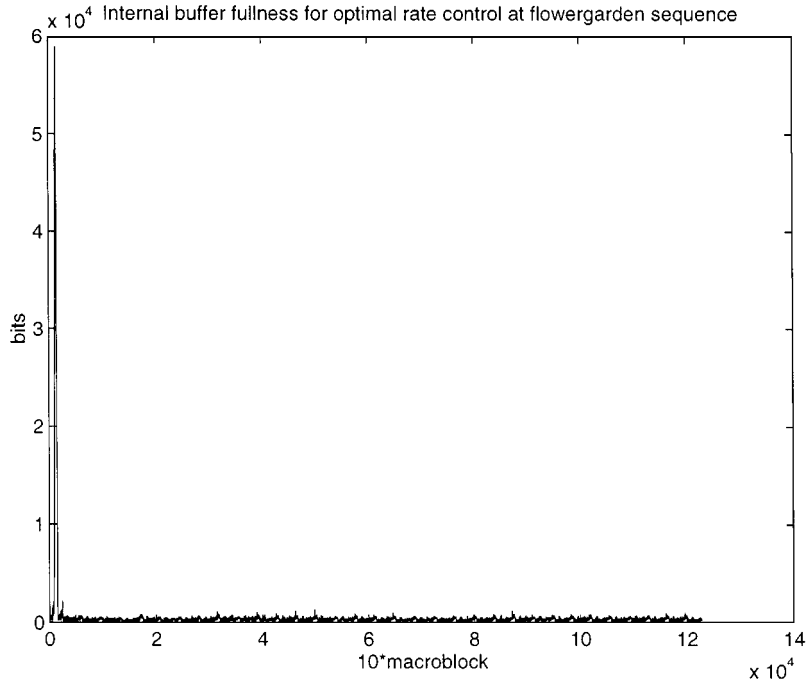


Figure 7. Internal buffer fullness of the optimal rate control algorithm for the *Flowergarden* picture sequence.

circuits to calculate the distortion at different quantization stepsizes. A_{ext} denotes the area of the additional hardware. The additional VLC unit only needs to produce the total bits of the codewords; it does not have to produce the bit streams. Thus, we use adder and divider to implement the additional VLC and quantization units. They are implemented by a dedicated structure to reduce the silicon area. The statistical analysis of the dedicated architectures (2 mm^2/GOP) for video codec [14] is used to estimate the area of these units. The calculated coded bits and distortion for various quantization stepsizes are stored in an on-chip buffer. In this paper, the on-chip buffer is classified into the additional hardware part, as listed in Table 4. The n_λ value in this table turns out to be 8, which is obtained through the simulation of the Football picture sequence.

Table 4 summarizes the silicon area needed for the major components in various rate control algorithms. In our approximation, the *mul* and *div* processing elements requires roughly 7 [14] and 16 [15] times the area of the *add* element. Thus, a *log* operation is equivalent to 60 *add* operations by using the third order Taylor series expansion. In this table, an area estimation model of two-port memory proposed by Chang [16] is adopted

Table 4. Estimated silicon area for various rate control algorithms.

Algorithm	Test Model 5			Budget planning				Optimal bit allocation			
N_{op}	add	mul	div	add	mul	div	log	add	mul	div	log
(MIPS)	20.8	10.4	0.04	31.2	0.12	0.16	0.041	20.7	10.4	$\simeq 0$	$\simeq 0$
A_{op} (mm^2)		9.4			3.7					9.4	
A_{ibuf} (mm^2)		5.3			0.29					54.3	
N_{ext}								add		div	buf
(MIPS)		—			—			995.3		1492.9	691200
A_{ext} (mm^2)		0			0					230.6	
A_{total} (mm^2)		14.7			4.0					294.3	

to estimate the buffer silicon area. To simplify the analysis, we only use the buffer size in estimating the silicon area and neglect the access time requirement. The last row of this table is the total silicon area A_{total} which is the combination of the processing unit, the internal output buffer, and the additional hardware unit. It is interesting to see that the area of the internal output buffer plays an important role in the total silicon area. From Table 4, we find that the silicon area of the optimal bit allocation algorithm is approximately 75 times larger than that of the budget planning algorithm, and the TM5 silicon area is about 20 times larger.

Furthermore, the bit allocation unit is also a dominant factor in the entire MPEG encoder chip area. For example, if the optimal bit allocation algorithm is used, its estimated silicon area can be larger than the area of all the other units combined [3]. Even in the case of the simple TM5 algorithm, it takes about the same area of the DCT unit, which is about 10% of the entire chip [3].

4. Picture Quality

Different rate control algorithms produce different image quality. Although peak signal-to-noise ratio (PSNR) is not a precise measure for subjective image quality judgment, it can still be used as a rough picture quality indicator. The PSNR is defined as the ratio of the peak signal power (255^2) to the mean square coded pixel errors. In this simulation, the three-step search algorithm is used for reducing the computing time and the search range is 47 for P-pictures and 15 for B-pictures. For a target bitrate of 5 Mbps, the encoded bits per picture for different rate control algorithms on the CCIR 601 image sequences (*Football* and *Flowergarden*) are shown in Figs. 8 and 9. Notice that the frame bits are identical for both TM5 and the optimal algorithm because the same frame bit allocation scheme is used in both cases. The PSNR performance of each image sequence is shown in Figs. 10 and 11. It is clear that the optimal rate control algorithm outperforms all the other algorithms. The adaptive budget planning algorithm is lower by roughly 1.5 dB in PSNR. TM5 is somewhere in between. Similar experiments are conducted on the Gaussian test picture sequence, and Figs. 12 and 13 show the PSNR and the coded bits. Unless there is a significant disadvantage in hardware cost, the optimal rate control algorithm seems to be the best candidate from the PSNR performance viewpoint. However, the objective PSNR performance differences among various rate control algorithms are not very significant. Hence, it is

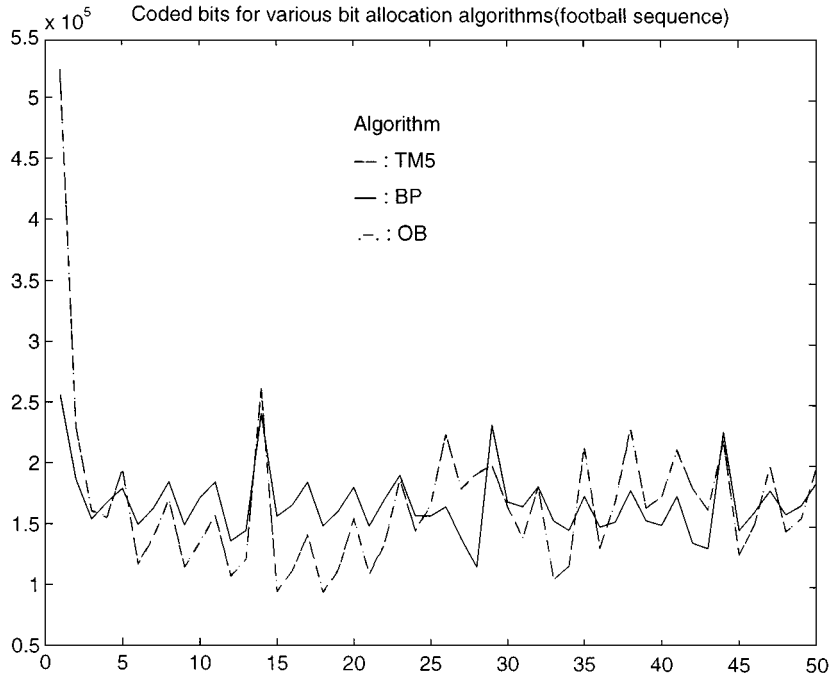


Figure 8. Frame bits allocation for various rate control algorithms on the *Football* sequence.

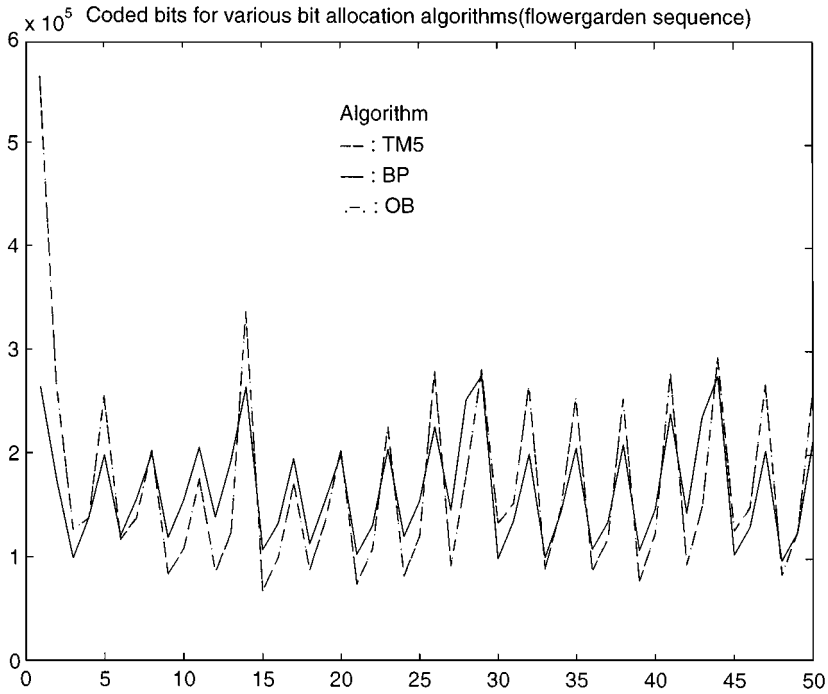


Figure 9. Frame bits allocation for various rate control algorithms on the *Flowergarden* sequence.

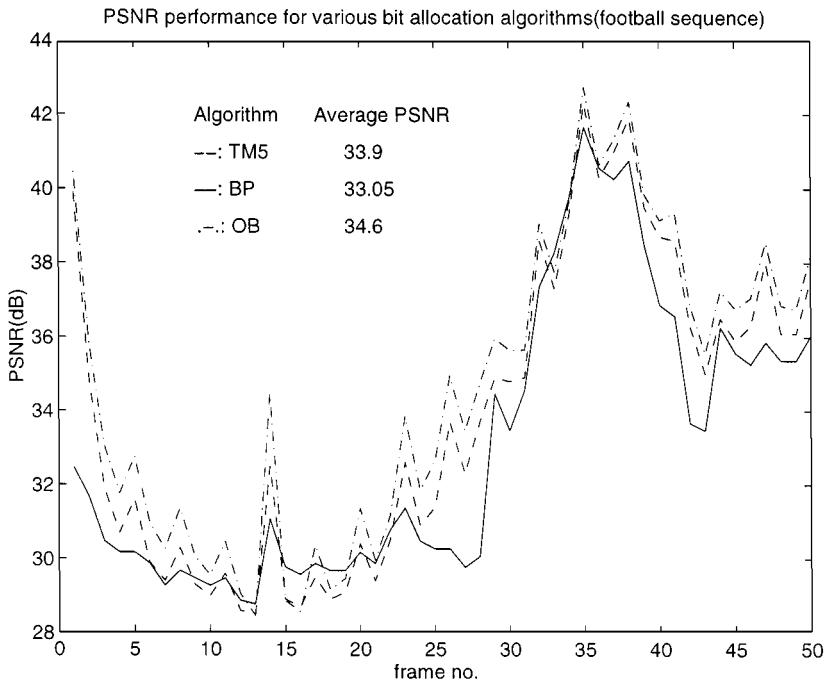


Figure 10. PSNR performance for various rate control algorithms on the *Football* sequence.

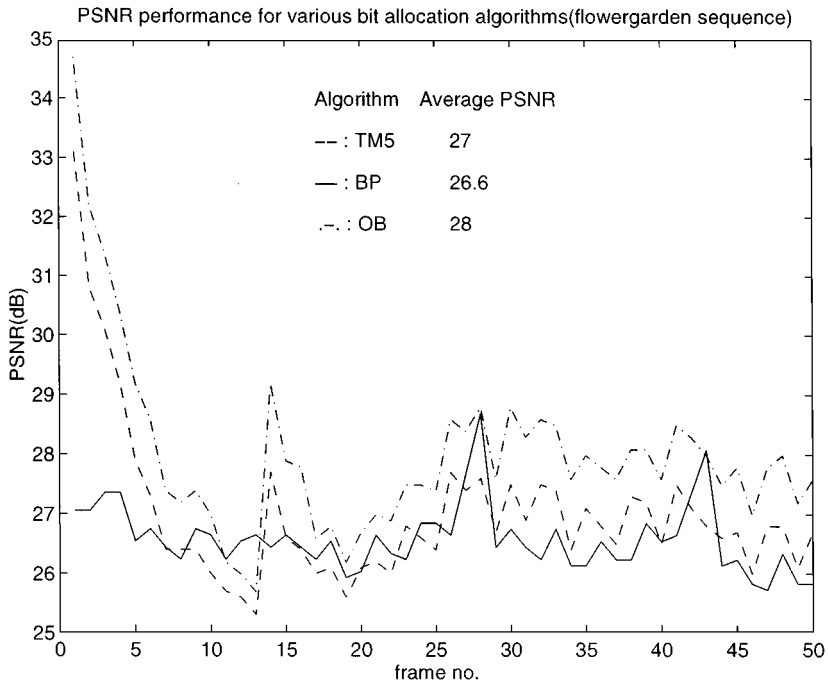


Figure 11. PSNR performance for various rate control algorithms on the *Flowergarden* sequence.

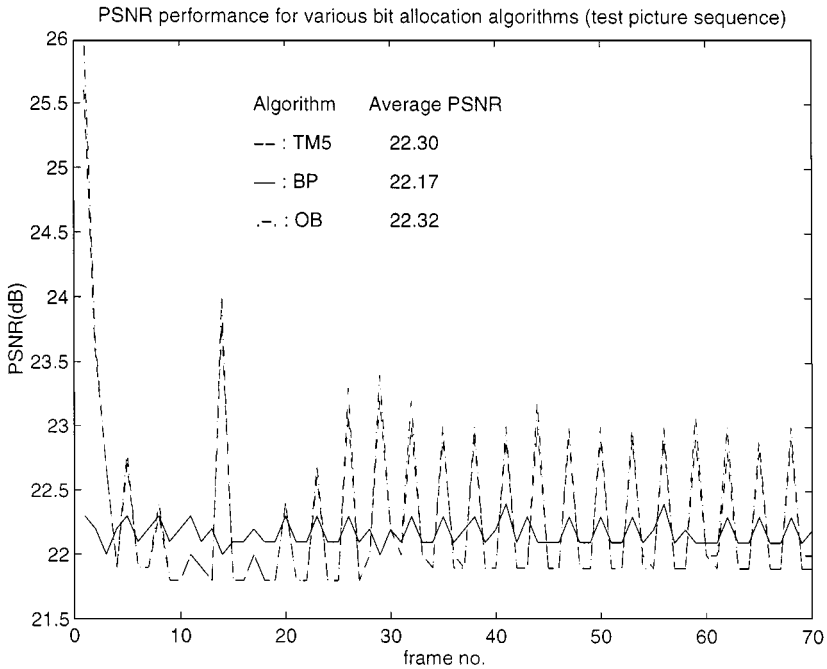


Figure 12. PSNR performance for various rate control algorithms on the *Gaussian* sequence.

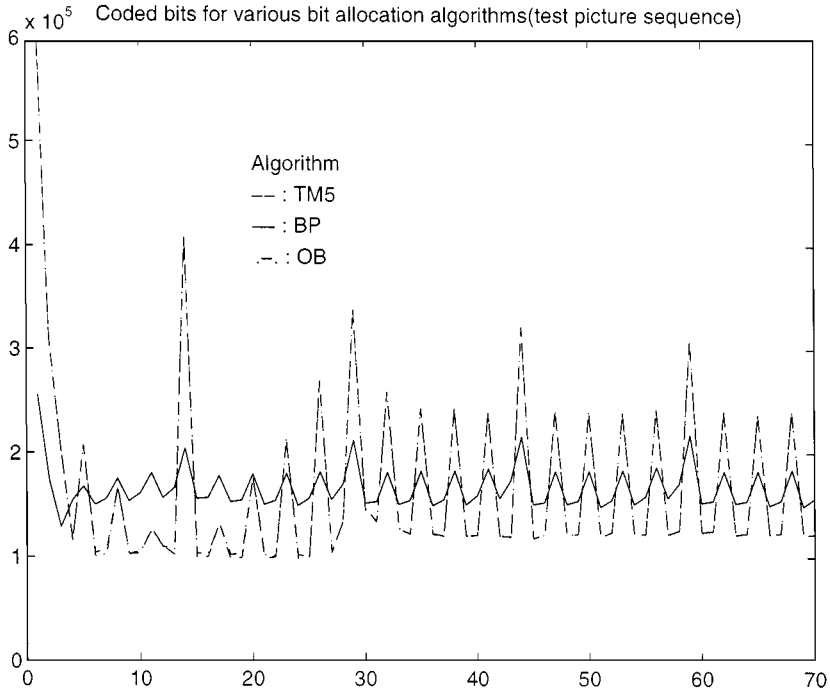


Figure 13. Frame bits allocation for various rate control algorithms on the *Gaussian* sequence.

worthwhile looking into the subjective visual quality. In the case of *Football* and *Flowergarden* sequences, all the coded picture subjective qualities are at a par. But there exists some noticeable differences on the difficult Gaussian test sequence. The center of the 10th frame of the Gaussian sequence is shown in Figs. 14–16. Subjectively, the adaptive budget planning algorithm has the best visual quality. Since the optimal rate control algorithm spends too many bits on the Gaussian noise background, it has the lowest subjective quality on the center picture. Figure 17 shows the PSNR values of the *Gaussian* sequence without the surrounding noise region. It is clear that the adaptive budget planning algorithm has the best performance. This is not surprising because in our budget planning algorithm, the macroblock bits assignment formula includes a log operator for achieving a more uniform visual quality at the cost of a lower overall PSNR value. The optimal bit allocation algorithm can also add a visual-dependent weighting function to its cost function to improve the subjective quality. In summary, the optimal rate control algorithm may act as the upper bound for the numerical PSNR performance, but in general, all three algorithms have quite close PSNR values. The budget planning algorithm has a somewhat better subjective quality because it includes a visual criterion in bits assignment.

5. Conclusion

The purpose of this study is not to propose a VLSI architecture for implementing a specific rate control algorithm but to evaluate various rate control strategies from the viewpoints of both VLSI design and coding performance. Three representative types of rate control algorithms are evaluated. A distinct feature in our study is to include the internal output buffer into the silicon area. In this paper, we found that the rate control algorithm plays an important role in the video encoder design. In addition to the rate-distortion performance, we should also consider the hardware implementation issue in designing a good rate control algorithm.

Extending previous studies, we use an adaptive piecewise linear bit model in the budget planning rate control algorithm to allocate the frame bits. Our experiments show that our simple budget planning rate control algorithm has a significant advantage in hardware cost while maintaining a comparable rate-distortion performance. The optimal rate control algorithm has its

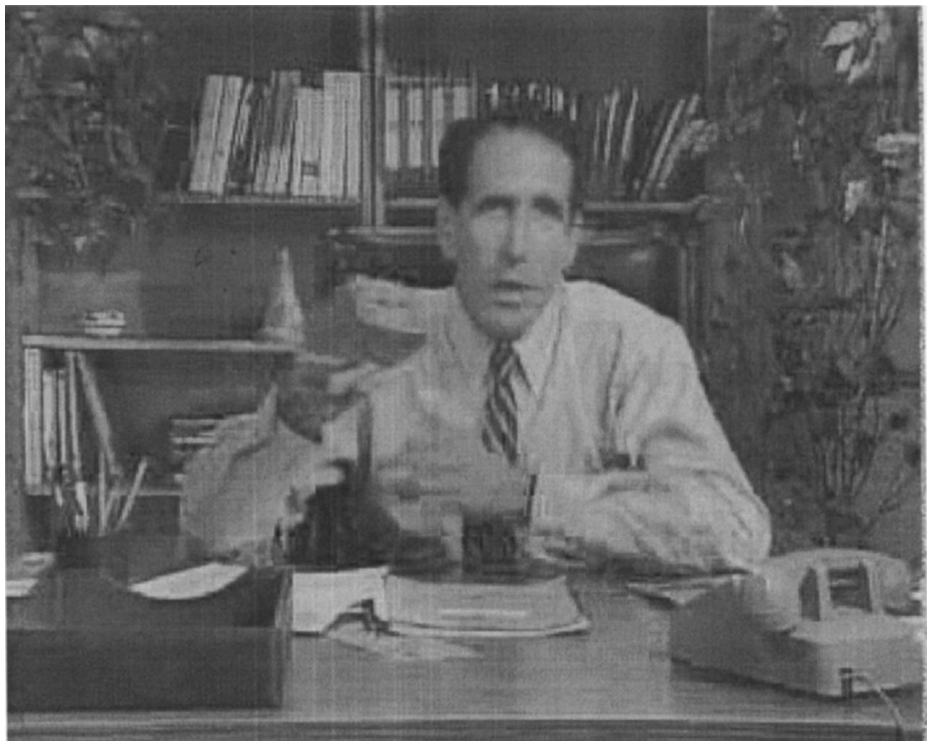


Figure 14. The coded 10th frame using *TM5*.

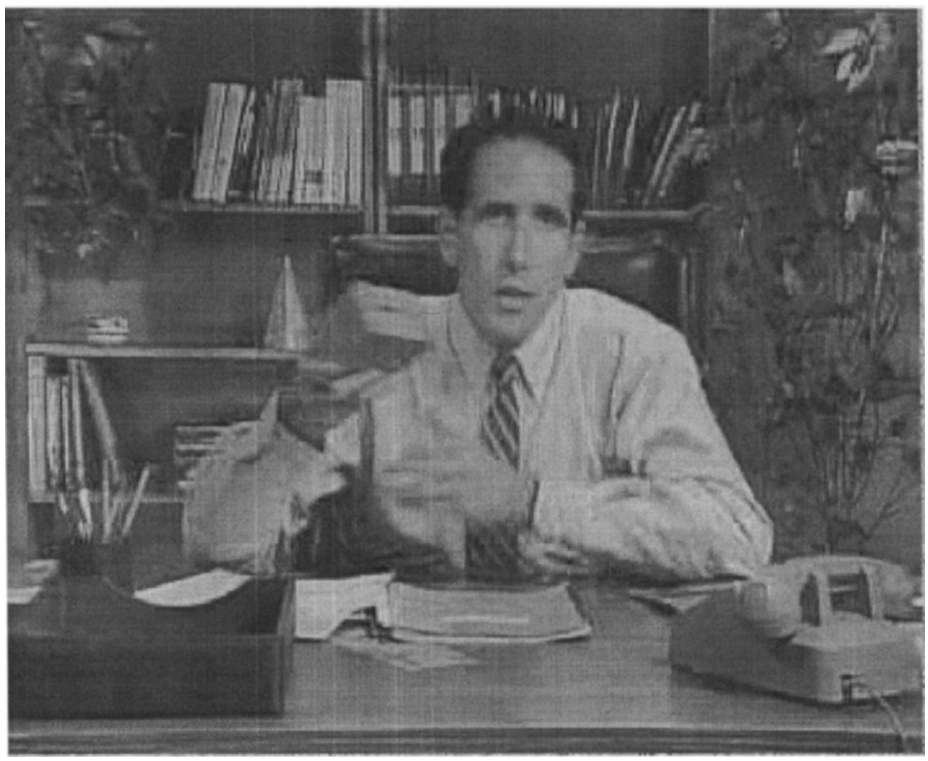


Figure 15. The coded 10th frame using the *adaptive budget planning* scheme.

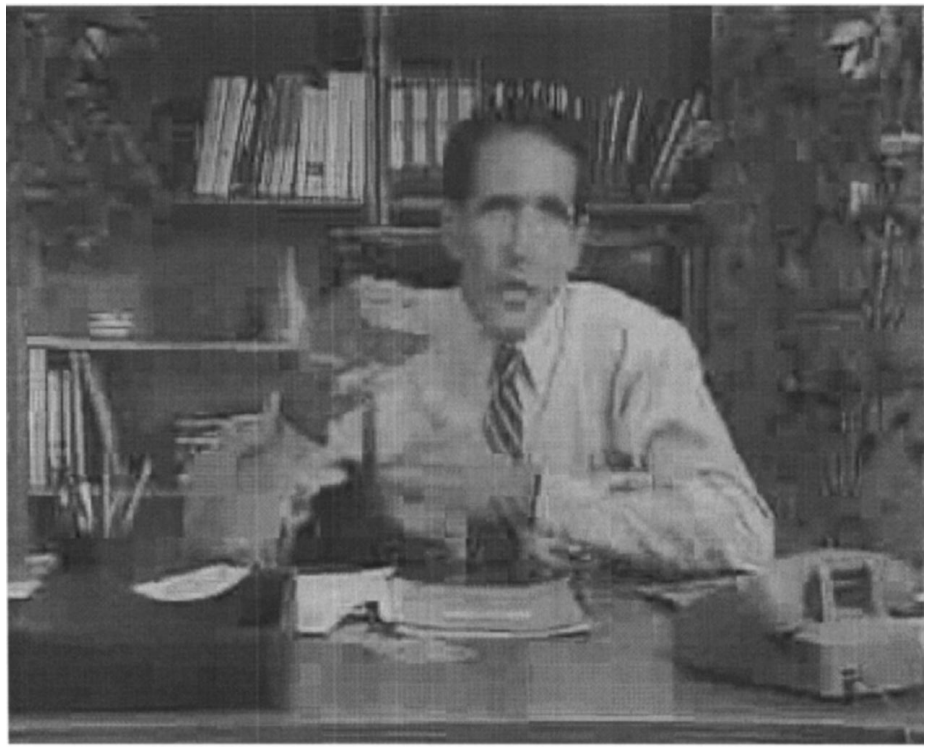


Figure 16. The coded 10th frame using the *optimal rate-distortion* scheme.

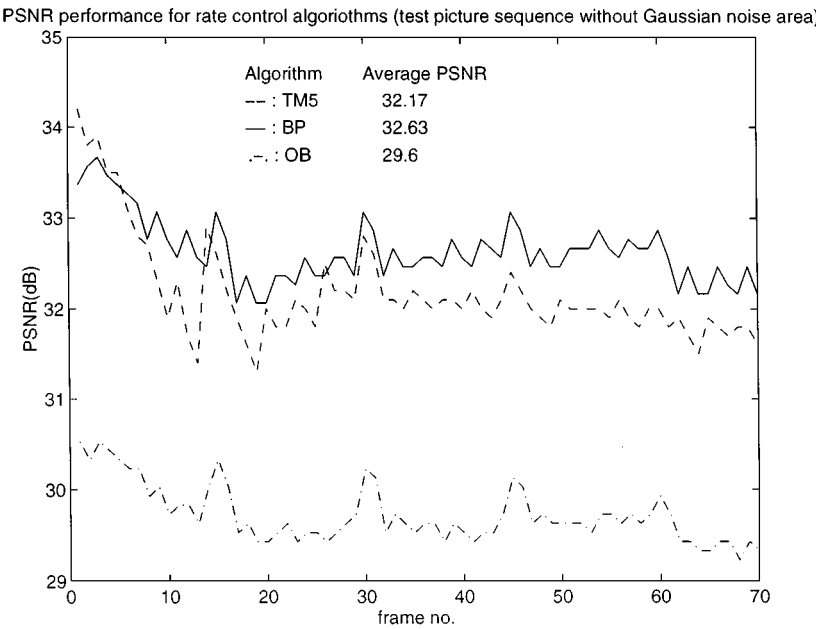


Figure 17. The PSNR of the center (*salesman*) of the (*Gaussian*) sequence.

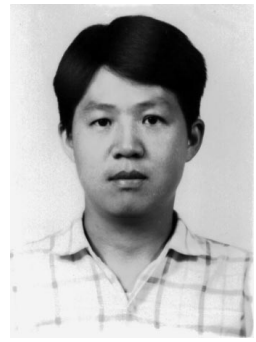
advantage in PSNR performance but requires a much higher hardware cost. Our analysis in this paper should be able to provide useful guidelines to system designers in choosing a suitable high-level rate control algorithm for VLSI implementation.

Acknowledgment

This work was supported by National Science Council of ROC under grant NSC86-2221-E-009-023.

References

1. K. Kucukcaker and A.C. Parker, "A methodology and design tools to support system-level VLSI design," Tech. Rep. Department of Electrical Engineering-Systems, University of Southern California, June 1994.
2. ISO/IEC JTCI/SC29/WG11 MPEG phase 2, Doc. NO400, "Test Model 5," April 1993.
3. S.-C. Cheng and H.-M. Hang, "The impact of encoding algorithms on MPEG VLSI implementation," *IEEE Int. Conf. on Circuits and System*, 1998, to appear.
4. S.-C. Cheng and H.-M. Hang, "A comparison of block-matching algorithms mapped to systolic-array implementation," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 7, pp. 741–757, Oct. 1997.
5. A. Puri and R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 1, No. 4, pp. 351–361, Dec. 1991.
6. CCITT, Working Party XV/4, Doc. 525, "Description of Ref. Model 8 (RM 8)," June 1989.
7. K.-W. Chow and Bede Liu, "Complexity based rate control for image encoder," *Int'l Conf. on Image Proc. '94*, Vol. 1, pp. 263–267, Nov. 1994.
8. W.-Y. Sun, H.-M. Hang, and C.-B. Fong, "Scene adaptive parameters selection for MPEG syntax based HDTV coding," *Int'l Workshop on HDTV '93*, Ottawa, Canada, Oct. 1993.
9. J.-B. Cheng and H.-M. Hang, "Adaptive piecewise linear bits estimation model for MPEG based video coding," *Visual Commun. and Image Represent.*, Vol. 8, No. 1, March 1997.
10. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. ASSP*, Vol. 36, No. 9, Sept. 1988.
11. K.M. Uz, J.M. Shapiro, and M. Czigler, "Optimal bit allocation in the presence of quantizer feedback," *Proceedings 1993 Internal Conference on Acoustics, Speech and Signal Proceeding*, Vol. 5, pp. 385–388, 1993.
12. K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with application to MPEG video coders," *Proceedings 1993 Internal Conference on Acoustics, Speech and Signal Proceeding*, Vol. 5, pp. 381–384, 1993.
13. W.-Y. Lee and J.-B. Ra, "Fast algorithm for optimal bit allocation in a rate distortion sense," *Electron. Lett.*, Vol. 32, No. 20, Sept. 1996.
14. P. Pirsch, N. Demassieux, and W. Gehrke, "VLSI architectures for video compression—A survey," *Proc. of the IEEE*, Vol. 83, No. 2, Feb. 1995.
15. Texas Instruments, 2547301-9721, rev. D, "TMS320C5x user's guide," Jan. 1993.
16. T.S. Chang, "On-chip memory module designs for video signal processing," Master thesis, Institute of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, ROC, June 1995.



Sheu-Chih Cheng received the B.S. degree in Electronics Engineering from National Taiwan Industrial Technology, Taipei, Taiwan, in 1989, and the M.S. degree from National Chiao Tung University in 1991. He is currently working toward Ph.D. degree in Electronics Engineering at National Chiao Tung University. His research interests are video coding and VLSI design for signal processing.

Hsueh-Ming Hang received the B.S. and M.S. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984. From 1984 to 1991, he was with AT&T Bell Laboratories, Holmdel, NJ. He joined the Electronics Engineering Department of National Chiao Tung University, Hsinchu, Taiwan, in December 1991. He was a conference co-chair of Symposium on Visual Communications and Image Processing (VCIP), 1993, and the Program Chair of the same conference in 1995. He guest co-edited two *Optical Engineering* special issues on Visual Communications and Image Processing in July 1991 and July 1993. He was an associate editor of *IEEE Transactions on Image Processing* from 1992 to 1994 and a co-editor of the book *Handbook of Visual Communications* (Academic Press, 1995). He is currently an associate editor of *IEEE Transactions on Circuits and Systems for Video Technology* and an editor of *Journal of Visual Communication and Image Representation*, Academic Press. He is a senior member of IEEE and a member of Sigma Xi.