In Proc. of Visual Communic. and Image Proc., VCIP '95, Taipei, Taiwan, May 1995

Gradient-based buffer control technique for MPEG

Liang-Jin Lin, Antonio Ortega and C.-C. Jay Kuo

Signal and Image Processing Institute and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, California 90089-2564

ABSTRACT

This paper proposes a new buffer-control technique for MPEG Video. The goal is to minimize the distortion while keeping the change in distortion between consecutive frames small. We formulate this goal as a constrained optimization problem and find the optimal solution using an iterative gradient search method. A preliminary experiment on a sub-GOP (4 frames, IBBP) shows that the solution is very close to the global optimal point. Further experiments on short video sequences show that, at the same bit rate and buffer constraint, the technique generates output sequences with smaller and more stable mean square error than other techniques, and strictly constant bit rate for every group of pictures. This technique should be particularly useful for off-line encoding or for one encoder/many decoders situations, where the higher computational cost required at the encoder can be afforded pays off by allowing higher quality sequences or smaller decoder buffers.

Keywords : video coding, rate control, buffer control, gradient search, steepest descent, coordinate descent

1 INTRODUCTION

Digital video compressed with the MPEG [1,2] standard has recently become increasingly popular. One key feature of MPEG, which also applies to other video encoding schemes, is that its encoding is much more complicated than decoding. It is well-suited for nonsymmetric applications, say, the data storage in a video-CD, where more computing power and processing time can be spent in encoding and the encoded data stream can be decoded with a relatively low cost. In video-CD decoding, data is read out from disk at a strictly constant speed thus making necessary a decoder buffer so that the video stream can be decoded and played back synchronously. The size of the buffer is an important component of the cost of the decoder and this motivates researchers to study effective buffer control schemes. Because the buffer control strategy affects only the complexity of the encoder, there is particular interest in strategies that might be complex but will reduce the required buffer size for a given video quality or increase the quality for a given buffer size. Beside applications in video-CD and CD-ROM, buffer control is also useful for video communication over constant-bit-rate (CBR) channels and for traffic shaping in variable-bit-rate (VBR) channels such as ATM networks.

Most current buffer control methods use buffer occupancy to determine the quantization scale for the next macroblock, slice, or frame by a direct mapping [3,4]. Other works have proposed modeling for the encoder bit rate in order to improve the control algorithm [5]. In both cases only the rate, not the distortion, is taken into account in the algorithm. Alternatively, the buffer control problem can be formulated as a delayed decision

constrained optimization problem and solved using dynamic programming [6] or Lagrangian techniques [7–9]. These approaches require additional encoding delay and complexity but provide optimal results in a rate distortion sense.

In this paper, we follow a forward adaptive optimization to develop a new buffer control technique based on iterative gradient search to reduce the computation complexity. In Section 2, we first formulate the buffer control as a constrained nonlinear optimization problem. In Section 3, we approximate the solution by introducing a penalty functions and using gradient search methods. In Section 4, we present some results. We first use a short GOP sequence to demonstrate that the performance is close to the optimal point achievable by exhaustive search. We also apply our technique to encode short video sequences and compare our performance with that achieved by MPEG's Test Model 5. Finally, conclusions and future extensions of this technique are discussed in Section 5. Note that although we provide experimental results for an MPEG encoder, the algorithms presented here are applicable to other video encoding scenarios as well.

2 PROBLEM FORMULATION

In MPEG, a video sequence is divided into Groups of Pictures (GOPs) with size equal to N frames. Typical values of N are 12 for a 25 fps PAL video, or 15 for 30 fps NTSC video. Each GOP can be decoded independently, so it can be considered as an access unit in MPEG stream. A strictly constant number of bits for each GOP is needed for recording on a digital tape recorder [10]. It also allows speeding up the searching and indexing process for the video stream in CD-ROM. The goal of our approach is to maintain constant bit rate for every GOP, keep the buffer occupancy within the constraints, while minimizing the distortion.

The number of bits generated by the MPEG encoder is determined by the the quantization scale, MQUANT, which can be changed over different macro blocks. In our formulation, the quantization scale, denoted as q, is kept constant over an entire frame. Note that performance would be improved if different MQUANT's were used for each block. Since our method applies to choosing one among a finite set of operating points, it would be easy to extend it to the case where the operating points are defined by their various "quality levels", with adaptive MQUANT allocations performed for each frame and quality level, as in [11]. In this work, we only use constant quantization for each frame to demonstrate the performance of our technique. Under this assumption, the buffer control problem is to assign the quantization scale q_i for the ith frame in a GOP such that the overall quality, measured by a pre-defined cost function, is optimized. Let $\mathbf{q} = (q_0, q_1, \ldots, q_{N-1})^T$ be the quantization choices for the frames in a GOP. When the quantization scales are set to $\mathbf{q}$, we define the code length and mean square error of frame i as the rate and distortion functions, denoted by $r(i, \mathbf{q})$ and $d(i, \mathbf{q})$, respectively. Both $r(i, \mathbf{q})$ and $d(i, \mathbf{q})$ are actually measured during the encoding process. By using a vector expression for $\mathbf{q}$, we are taking into account the "dependency" of the problem, i.e., the distortion/rate trade off for predicted/interpolated frames depends on the frames that were used to generate the prediction [8]. The buffer occupancy after frame i has been coded is then (assuming underflow does not occur):

$$b(i, \mathbf{q}) = b_0 + \sum_{k=0}^{i} r(k, \mathbf{q}) - iR \tag{1}$$

where b_0 is the buffer occupancy at the start of a GOP (usually zero), and R is the channel bit-rate in bits per frame. In order to allocate a constant number of bits for each GOP, the final buffer occupancy $b(N-1, \mathbf{q})$ should be less than zero and as close to zero as possible. At the end of each GOP, stuff bits are appended to make buffer occupancy zero.

We define the cost function as

$$J(\mathbf{q}) = D(\mathbf{q}) + wE(\mathbf{q}) \tag{2}$$

where $D(\mathbf{q})$ represents the average distortion and is defined as

$$D(\mathbf{q}) = \frac{1}{N}\sum_{i=0}^{N-1} d(i,\mathbf{q}) \tag{3}$$

$E(\mathbf{q})$ is the average squared difference in distortion between consecutive frames:

$$E(\mathbf{q}) = \frac{1}{N}\sum_{i=0}^{N-1} [d(i,\mathbf{q}) - d(i-1,\mathbf{q})]^2 \tag{4}$$

and w is the weighting coefficient between $D(\mathbf{q})$ and $E(\mathbf{q})$. The purpose of $E(\mathbf{q})$ in the cost function is to minimize the abrupt changes in quality and avoid the "flicker problems". Note that, although we choose MSE as the distortion measure in this paper, it is also possible to use weighted MSE instead. For instance, MSE weighted by an "activity" measure as in [3] could easily be used.

The problem can now be formulated as that of finding $\mathbf{q}^\star$ such that:

$$\mathbf{q}^\star = \arg\min_{\mathbf{q}} J(\mathbf{q}) \tag{5}$$

subject to

$$q_i \in \{1, 2, \ldots, 31\}\,, \quad i = 0 \ldots N-1 \tag{6}$$

$$0 \le b(i,\mathbf{q}) \le b_{max}\,, \quad i = 0 \ldots N-2 \tag{7}$$

$$b(N-1,\mathbf{q}) \le 0 \tag{8}$$

where b_{max} is the prescribed maximum buffer size. Note that in this work we choose to consider underflow as a constraint (see (7)). It would also be possible to relax the underflow constraint by padding stuff bits when it occurs, and letting the buffer occupancy in (1) be $b(i,\mathbf{q}) = \max\{0, b(i-1,\mathbf{q} + r(i,\mathbf{q}) - R)\}$. In (8), we force the final buffer occupancy to be less than or equal to zero, and pad stuff bits at the end of GOP to ensure all GOPs have the same number of bits.

3 SOLUTION USING GRADIENT SEARCH TECHNIQUES

The optimization problem formulated in the previous section is an integer programming problem with a nonlinear cost function and nonlinear constraints. These characteristics make the optimization problem difficult and make the need for efficient solutions all the more pressing. One approach that has been proved to be useful in this context is dynamic programming techniques [6]. Although the true global optimal solution can be obtained from this approach, the computation cost is high. In this paper, we do not intend to obtain the global optimal solution. Instead, we are only looking for reasonable suboptimal solution with reduced computation cost.

In order to reduce reduce the complexity, our first approximation is to change the integer-valued variable into continuous one, so that many optimization techniques defined in continuous domain can be applied.

3.1 Penalty functions

The constraints of (7) and (8) can be taken into account by adding penalty functions to the cost, $J(\mathbf{q})$. The penalty functions are defined as

$$P_i(\mathbf{q}) = \{\min[0, b(i,\mathbf{q})] + \max[0, (b(i,\mathbf{q}) - b_{max})]\}^2 \quad (9)$$

$$Q(\mathbf{q}) = \{\max[0, b(N-1,\mathbf{q})]\}^2 \quad (10)$$

The new cost function is

$$\phi(\mathbf{q}, c) = J(\mathbf{q}) + c\left(\sum_{i=0}^{N-2} P_i(\mathbf{q}) + Q(\mathbf{q})\right) \quad (11)$$

where the parameter c determines the amount of the penalty. The original problem can be approximated by iteratively solving the unconstrained problem of minimizing $\phi(\mathbf{q}, c)$ as $c \to \infty$.

3.2 Iterative gradient search

In order to solve the unconstrained problem efficiently, we make the assumption that the cost function is smooth. Because the encoding process is a nonlinear operation, it is not easy to justify the smoothness of the cost function. In a preliminary experiment, we encode two frames of Miss America sequence with every possible setting of quantization scales, and calculate the cost function. The surface plot of the function is shown in Figure 1. We observe a smooth surface for the cost function which indicates that iterative gradient methods can be used. In a real situation, where several frames are considered, the function may not be perfectly smooth, but an only slightly suboptimal point may still be reachable. Experiments presented in Section 4.1 verify that gradient search methods provide a good approximation.

There are several iterative gradient search algorithms available for our problem [12,13]. The first one that we have considered is *cyclic coordinate descent method*, also known as *alternating variables method*. This method minimizes the cost function with respect to one coordinate component q_i at a time, by solving

$$\min_{q_i} \phi(q_0, q_1, \ldots, q_{N-1}) \quad (12)$$

The minimization is with respect to q_0 first, then q_1, and so on, up to q_{N-1}. The process is then repeated starting with q_0 again, until convergence is achieved. The advantage of this method is its simplicity. While this method has the drawback of not having a strong theoretical guarantee for convergence [13], we consider it in order to avoid the costly computation for the gradient.

The second method that we have applied is the *steepest descent method*. In this method, we use the negative direction of the gradient vector $\nabla\phi(\mathbf{q})^T$ as the search direction, and update the vector $\mathbf{q}$ by the following

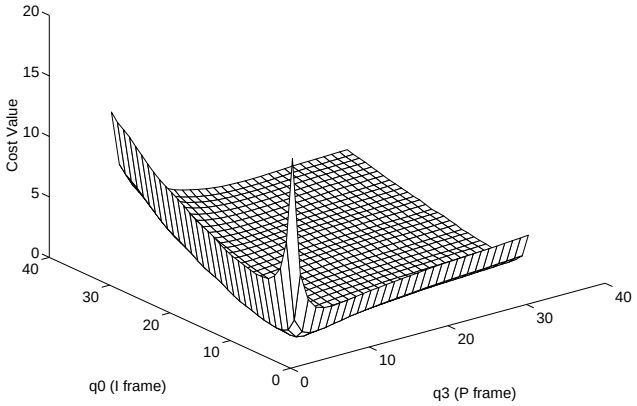

Figure 1: Typical cost function in two dimensional case

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \alpha_k \nabla\phi(\mathbf{q}_k)^T \tag{13}$$

where α_k is a nonnegative scalar value. There are two ways of choosing the value of α_k: either use a fixed value, or use the value that minimizes the function $\varphi(\alpha) = \phi\left(\mathbf{q}_k - \alpha\nabla\phi(\mathbf{q}_k)^T\right)$, which can be done by a line search algorithm.

Besides the above two approaches, there are several other methods that aim at improving the convergence rate, including the conjugate gradient and quasi-Newton method. These methods are generally preferred for the cost functions that include a penalty (as in our formulation), because convergence is usually slow when the penalty parameter c is large [12]. Unfortunately, those methods require the matrix of second derivative, or Hessian, either by direct computation or by indirect estimation, which is, in our case, costly in computation. Also, because we are not dealing with continuous functions, the error in approximating our discrete set of points is subject to be amplified during the computation of the second derivative, which might cause the algorithm to be unstable. So, we do not use these advanced algorithms at the current stage. The feasibility of those methods is left for future research.

3.3 Integer approximation

In our formulation, the variable $\mathbf{q}$ is only defined in the discrete integer grid. For the cyclic coordinate method, this is not a problem, because the search direction is always parallel to the coordinate axis and always falls on the grid. But this is not true for the steepest descent method, where the search direction can be arbitrary. So, it is not possible to perform an exact line search. Instead, we create a search path consisting of the points that are closest to the line, as indicated in Figure 2 for the two dimensional case. Those big points can be efficiently derived by the line drawing algorithm used in raster computer graphics [14]. Another problem is on the calculation of the gradient vector, where the derivative is required. In this paper, we use the first-order difference to approximate the derivative.

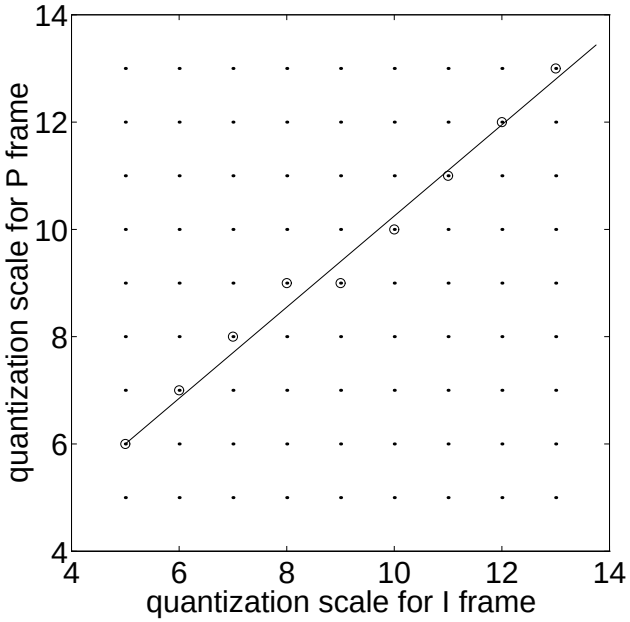

Figure 2: *Discrete line search in 2-D case* The solid line indicates the path along negative gradient direction. The actual search path is along the points indicated by the circled dot.

3.4 Initialization

One problem of the gradient search is that, it only converges to a local minimum solution. Hence the initial guess is an important factor for the performance, affecting both the achievable local minimum and the rate of convergence. Before coding the first GOP, or at a scene change, we have no prior knowledge of the sequence and thus the first initial guess will be based on "typical" quantization scales. Perhaps the best way to encode the first GOP would be to use the dynamic programming technique [6] to obtain the true global optimum solution, but in this paper, we just set the initial guess to a fixed set of typical values, for example, 8 for I frame, 10 for P frames, and 14 for B frames. For the succeeding GOPs, we can take advantage of the similarity between adjacent GOPs, and use the solution for the previous GOP as initialization for the current one.

4 EXPERIMENTAL RESULTS

Our initial goal is to compare several approaches for the iterative search process. Our objective is twofold: to study their relative performance and to find out how much they deviate from the optimal solution. We thus consider an experiment on a sub-GOP, which is a set of four frames with assigned as I, B, B, P respectively for each frame. Because in this case the optimal solution can be derived by exhaustive search, we can compare each of our methods with the optimal solution. We then apply the algorithm to encode short video clips.

4.1 Preliminary experiments

In this experiments, we choose a set of four frames from Miss America video sequence, and then encode it by an MPEG encoder with all possible combinations of quantization settings to measure the actual distortions

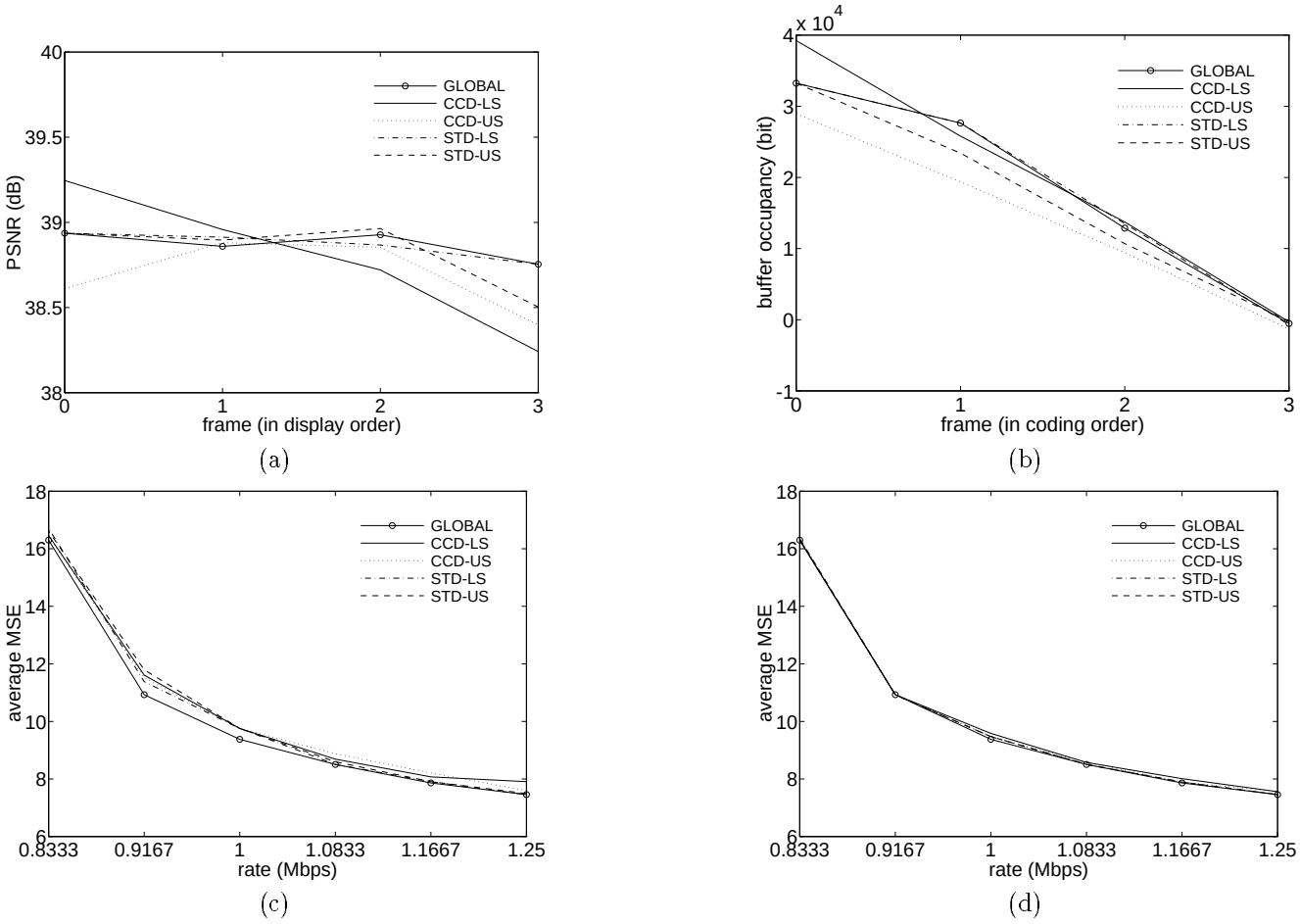

Figure 3: *Four frame experiment* (a) PSNR at bit rate 1.0833 Mbps, (b) buffer occupancy at bit rate 1.0833 Mbps, (c) rate-distortion curve. (a), (b), (c) are the results for single-pass penalty method, where c is fixed to 0.05. (d) rate-distortion curve for iterative penalty method, where c is iterated from 10^{-10}, multiplied by 10 after each iteration, until the solution converges and the constraints are satisfied. In each figure, *GLOBAL*: exhaustive search to obtain the global optimum solution; *CCD-LS*: cyclic coordinate descent with line search strategy; *CCD-US*: cyclic coordinate descent with unit stepping strategy; *STD-LS*: steepest descent with line search strategy, and *STD-US*: steepest descent with unit stepping strategy.

and rates in all cases. A modified version of the MPEG-1 Encoder version 1.3 from U. C. Berkeley is used for the encoding [15]. To simplify the experiment, we set the weighting factor w in (2) to be the ratio between the standard deviation of $D(\mathbf{q})$ and $E(\mathbf{q})$.

The following five search strategies are considered.

- *GLOBAL* Global optimum solution: Exhaustive search over all combinations of quantization settings is used to obtain the optimal solution and serves as a benchmark for the other methods.
- *CCD-LS* Cyclic coordinate descent with line search strategy: In each iteration, to approximate the solution in (12), a simple line search is used to find the first local minimum.
- *CCD-US* Cyclic coordinate descent with unit stepping strategy: In each iteration, instead of trying to find the minimum for each coordinate variable, we only allow changes of $+1$, 0, and -1 in the descent direction

	$c = 0.05$			iterated on c		
Method	Relative Error	PSNR (dB)	Complexity	Relative Error	PSNR (dB)	Complexity
GLOBAL		38.3024	1.5×10^4		38.3024	1.5×10^4
CCD-LS	0.04234	38.1580	16.29	0.01062	38.2558	68.46
CCD-US	0.03671	38.1529	14.08	0.00685	38.2738	50.08
STD-LS	0.01922	38.2198	15.46	0.00163	38.2893	81.25
STD-US	0.02851	38.1884	20.13	0.00163	38.2874	117.25

Table 1: Four frame experimental result. The *relative error* is the relative difference of the cost value between the specific search method and the global solution. The *complexity* is the total computation requirement relative to one-pass encoding (where 4 frames are coded). All the values are the average over the results from the six test cases, at bit rate 0.8333, 0.9167, 1.0, 1.0833, 1.1667, and 1.25 Mbps, respectively.

for each coordinate variable.

- *STD-LS* Steepest descent with line search strategy: In each iteration, the first order difference for each coordinate variable is used to approximate the gradient vector, and then the discrete line search method shown in Figure 2 is used to locate the first local minimum.
- *STD-US* Steepest coordinate descent with unit stepping strategy: In each iteration, instead of trying to find the minimum along the negative gradient direction, we go one step forward to the nearest point along the descent direction.

For the penalty parameter c in (11), the following two choices are used in our experiment: either use a fixed value at $c = 0.05$, or use an iterative strategy by initially setting $c = 10^{-10}$, solve the optimization problem in each iteration, multiply c by 10 after each iteration, until the solution converges and all the constraints are satisfied.

The results for the above procedures are running over several different rate settings are summarized and shown in Figure 3 and Table 1. The computational complexities are also presented in the table. Complexity is normalized by the complexity of one-pass encoding (which requires four frames to be encoded). Thus a complexity of n means that $4n$ frames would have to be encoded. We have the following conclusion:

- A good approximation to the optimal solution can be obtained at a fraction of the complexity. Compared to the exhaustive search, the reduction in complexity is in the order of $1,000$!.
- The method that iterates over the penalty parameter c has much better approximation to the optimal solution, at the expense of a five-fold increase in the computational complexity over the method that uses only one fixed c.
- Among the search methods, steepest descent with line search strategy has the best performance.
- The computation complexity seems to be at least 15 times higher than the single pass encoding algorithm. In actual implementation, it can be lower than this value. For example, the costly motion estimations can be done by using the original frames and only have to be done once during the whole encoding process. Furthermore, we can take advantage of the fact that there is no frame dependency on the quantization setting of the B frames. Finally, the search will be sped up significantly by using the result of previous GOPs as an initial guess for the current one.

As a trade off between the performance and computation complexity, in the following video encoding simulations, we choose the steepest descent with line search strategy, and use one fixed penalty parameter at $c = 0.05$ for the cost function.

4.2 MPEG encoding

Football at 1.152 Mbps							
	Test Model 5		Steepest Descent				
GOP Size	12	6	6	6	6	6	12
Weighting Factor (w)	$\times$	$\times$	0	1	10^3	10^6	10^3
PSNR (dB)	30.6895	30.5728	31.1776	30.8761	30.8281	30.7809	30.9603
Complexity	1	1	28.8	52.2	56.8	51.1	339.6
Complexity for Last GOP	1	1	22	15	26	22	301

Miss America at 1.152 Mbps					
	Test Model 5		Steepest Descent		
GOP Size	12	6	6	6	6
Weighting Factor (w)	$\times$	$\times$	0	10^3	10^6
PSNR (dB)	41.4331	41.3376	42.1133	42.0891	42.1121
Complexity	1	1	18.1	15.9	15.0
Complexity for Last GOP	1	1	20	14	14

Miss America at 0.8 Mbps					
	Test Model 5		Steepest Descent		
GOP Size	12	6	6	6	6
Weighting Factor (w)	$\times$	$\times$	0	10^3	10^6
PSNR (dB)	40.8956	40.7740	41.4301	41.3819	41.3819
Complexity	1	1	25.5	31.7	31.7
Complexity for Last GOP	1	1	24	21	21

Table 2: MPEG encoding results.

In this part of the experiment, two video clips in SIF format are used: *Football* sequence, 352×240, 30 frames per second; and *Miss America* sequence, 352×288, 25 frames per second. For each video clip, 22 frames are used for the encoding. In our experiment, we use a program derived from the MPEG-2 encoder version 1.1a, published by MPEG Software Simulation Group [16]. One major modification is that the motion estimation is done by using the original frames and is independent of the quantization setting. By doing so, we avoid the motion search in every iteration when using the steepest descent method and speed up the experiment. Although the modification will degrade the performance, the degradation is less then 0.1 dB in all our test cases. In all the experiments, the buffer size is set to 20×16384 bits. The original Test Model 5 rate control procedure that is included in the encoder is also used for comparison purposes.

The average PSNR and computation complexity for encoding the Football and Miss America sequence is presented in Table 2. As before, the complexity is normalized by that of a one-pass GOP encoding. Figure 4 (a), (b) shows the PSNR and buffer occupancy for Football sequence at 1.152 Mbps. The PSNR curve is presented in display order, while the buffer occupancy is in coding order. From the figure, we observe higher PSNR for steepest descent method. It is still better even when we compare GOP size 6 steepest descent method and GOP size 12 Test Model 5. Also, when the squared difference of MSE, $E(\mathbf{q})$, is introduced in the cost function, the PSNR curve becomes smoother, hence, results in a video that is more stable in quality. Figure 4 (c), (d) shows a similar result for Miss America sequence at 1.152 Mbps. The effect of weighting coefficient w is presented in Figure 5 (a), (b), and finally, a test case for steepest descent method with GOP size 12 is shown in Figure 5 (c), (d).

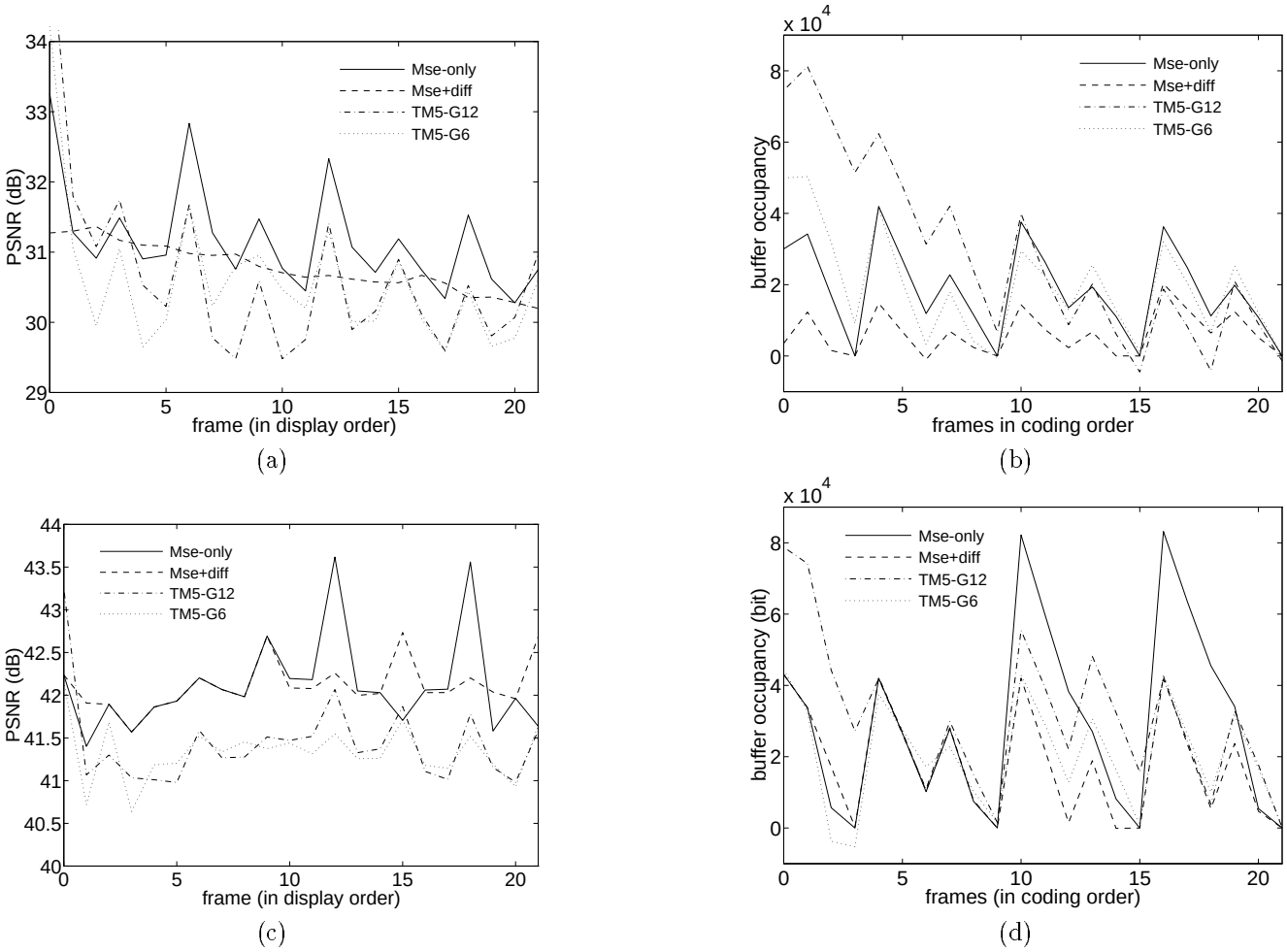

Figure 4: (a) PSNR, (b) buffer occupancy of Football sequence; (c) PSNR, (d) buffer occupancy of the Miss America sequence. The bit-rate is set to 1.152 Mbps for both sequence. In each figure, *Mse-only*: Steepest descent method with $w = 0$ and GOP size = 6; *MSE+diff*: Steepest descent method with $w = 10^6$ and GOP size = 6; *TM5-G12*: Test model 5 with GOP size = 12; *TM5-G6*: Test model 5 with GOP size = 6.

5 CONCLUSIONS AND FUTURE WORK

We have implemented and demonstrated the feasibility of using a gradient-based optimization algorithm for buffer control. By using this technique, we are able to achieve strictly constant bit rate per GOP, increase the overall quality (in MSE sense), while decreasing the variation of qualities between different frames. From the results that we have reached, we would like to emphasize the following

- In Test Model 5 algorithm, an adaptive quantization scheme is used to encode a frame. Compared to constant quantization, the adaptive quantization usually can have higher PSNR. From the experimental result, we observe that, even though our proposed algorithm uses a constant quantization scheme over an entire frame, the PSNR is still higher than that of Test Model 5. That means, we still have plenty of space for better PSNR, by adding the adaptive quantization into our scheme. In the future work, we plan to apply the technique of [11] to perform adaptive quantization within frames.

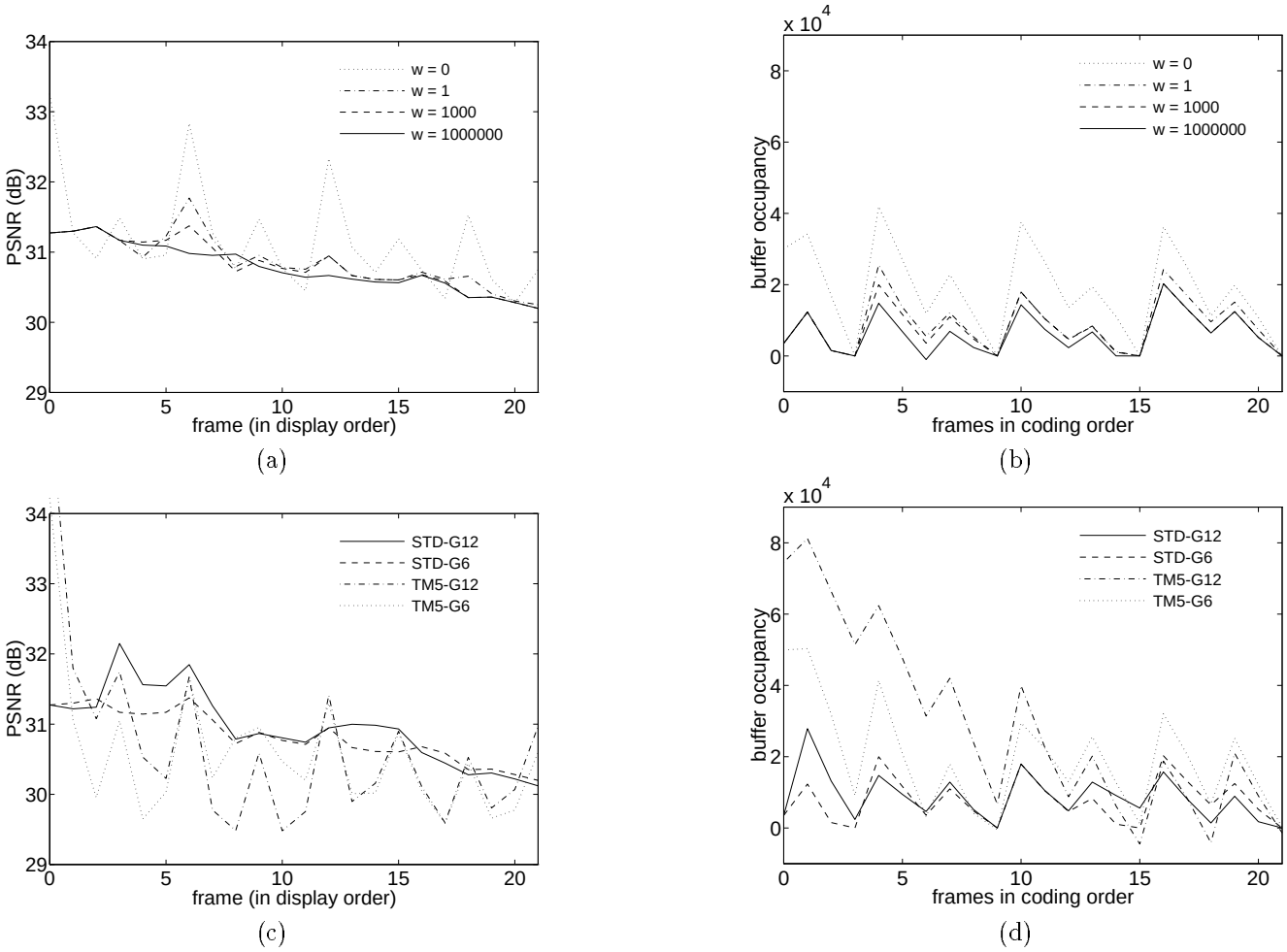

Figure 5: (a) PSNR, (b) buffer occupancy for Football sequence, over several different settings of w. These two figures show the effect of the weighting coefficient, w. (c) PSNR, (d) buffer occupancy for Football sequence, *STD-G12*: Steepest descent with GOP size = 12; *STD-G6*: Steepest descent with GOP size = 6. *TM5-G12*; Test model 5 with GOP size = 12; *TM5-G6*: Test model 5 with GOP size = 6.

- Regarding to the computational complexity, compared to Test Model 5, we can observe the complexity is 30 - 50 times higher for the first few GOPs, where the initial guess is far away from the optimum solution. In the follow up GOPs, because the solution in the previous GOP is used for the initial guess, the complexity is reduced to 15 to 25. The real complexity can be made smaller by taking out the costly motion estimation in the iteration loop. The inter-frame relationship can also be used to eliminate the unnecessary computations. These additional factors for complexity reduction will be taken into account in future work.
- We have observed that our technique using GOP size 6 already have better performance than Test Model 5 using GOP size 12. When we increase the GOP size 12 in our technique, the computation complexity becomes excessively high due to the slow convergence in higher dimensional vector space, while the increase in PSNR is not significant. In future work, we will modify the algorithm so that it can be applied to higher GOP size in more reasonable way, by taking into account the frame dependencies.

Although better quality (in MSE sense) can be achieved by the gradient technique, it comes at higher computation complexity. While this should not be a problem for off-line encoding, we also plan to use further

approximation for real time encoding. For example, we can introduce a mathematical model to estimate the rate and distortion of the next frame before actual coding it, based on the properties of previous frames that have already been coded. The same model also can be used to estimate the Hessian matrix so that more advance gradient iterations can be applied.

6 REFERENCES

[1] ISO CD11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s, Nov. 1991.

[2] Inform. Technology - Generic Coding of Moving Pictures and Associated Audio, ITU Draft Rec. H.262, ISO/IEC 13818-2, Mar. 1994.

[3] MPEG video simulation model three, ISO, coded representation of picture and audio information, 1990.

[4] C.-T. Chen and A. Wong. A self-governing rate buffer control strategy for pseudoconstant bit rate video coding. *IEEE Trans. on Image Proc.*, 2(1):50–59, Jan. 1993.

[5] J. Zdepsky, D. Raychaudhuri, and K. Joseph. Statistically based buffer control policies for constant rate transmission of compressed digital video. *IEEE Trans. on Comm.*, 39(6):947–957, June 1991.

[6] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal trellis-based buffered compression and fast approximation. *IEEE Trans. on Image Proc.*, 3(1):26–40, Jan. 1994.

[7] D. W. Lin, M.-H. Wang, and J.-J. Chen. Optimal delayed-coding of video sequences subject to a buffer-size constraint. In *Proc. of SPIE Visual Communications and Image Processing '93*, Cambridge, MA, Nov. 1993.

[8] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders. *IEEE Trans. on Image Proc.*, 3(5):533–545, Sept. 1994.

[9] J. Lee and B. W. Dickinson. Joint optimization of frame type selection and bit allocation for MPEG video encoders. In *Proc. of ICIP 94*, volume II, pages 962–966, Austin, Texas, 1994.

[10] S.-W. Wu and A. Gersho. Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV. *IEEE Trans. on Circuits and Sys. for Video Tech.*, 1(1):100–112, March 1991.

[11] A. Ortega and K. Ramchandran. Forward-adaptive quantization with optimal overhead cost for image and video coding with applications ot mpeg video coders. In *Proc. of IS&T/SPIE. Digital Video Compression '95*, San Jose, CA, Feb. 1995.

[12] D. G. Lueberger. *Linear and Nonlinear programming.* Addison-Wesley, 1984.

[13] R. Fletcher. *Practical Methods of Optimization.* Wiley, 1987.

[14] D. Hearn and M. P. Baker. *Computer Graphics.* Prentice-Hall, 1986.

[15] Berkeley Plateau Research group, MPEG-1 encoder version 1.3.
URL: `ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/mpeg/mpeg_encode-1.3.tar.Z`.

[16] MPEG Software Simulation Group, MPEG-2 Encoder version 1.1a.
URL: `ftp://ftp.netcom.com/pub/cfogg/mpeg2/mpeg2codec_v1.1.tar.gz`.