FAST PIECEWISE LINEAR APPROXIMATION OF RATE-DISTORTION FUNCTIONS FOR MPEG VIDEO

Chun-hang Choi* and Oscar C. Au**

Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: eechchoi@ee.ust.hk*, eeau@ee.ust.hk**

ABSTRACT

Rate Distortion based bit allocation algorithms were previously proposed to yield minimum distortion for a given bit rate in the framework of MPEG. However, they are impractical due to the huge computation required to generate the rate-distortion curve for each image block. In this paper, we propose a fast piecewise linear approximation of the rate distortion function that makes rate-distortion based bit allocation close to practical. By using the proposed fast recursive algorithm to compute selected points on the rate-distortion function and then apply linear interpolation, we show that the computation can be reduced by a factor of approximately 17. Simulation is performed in which rate distortion based bit allocation using bisection approach is applied to an MPEG-1 coder. A significant gain of 1.15dB in PSNR is found to be possible. But the proposed fast algorithm can only achieve a PSNR gain of 0.64dB suggesting that further work is needed.

Keywords : MPEG, video coding, rate distortion, bit allocation, bit control.

1. INTRODUCTION

In recent years, discrete cosine transform (DCT) based video coding with rate control feedback forms the basic framework of most of the current standards such as MPEG-1, MPEG-2, ITU-T H.261, and H.263. In these systems, each video frame is divided into blocks which are discrete cosine transformed, uniformly quantized and losslessly encoded with variable bit rate (VBR) coding. For the predictive coded blocks, motion compensation is applied before the DCT to reduce the temporal redundancy so as to achieve better compression. As the output video bitstream does not naturally has constant bit rate (CBR), adaptive scalar quantization is required to control the bit rate for many CBR applications such as CD-ROM storage and ATM transmission.

In these DCT-based video coding standards, the rate control and bit allocation methods are not explicitly specified. Bit allocation involves allocating different amount of bits to different blocks by selecting the mode of operation and the step size of the scalar quantizer for each of the coding blocks to achieve the best picture quality with a given amount of bits in a picture frame. Bit rate control, on the other hand, involves monitoring the buffer fullness, modeling the bitstream behaviour under different video context[1], and controlling the buffer occupancy at an acceptable and stable level by adjusting the step size of the scalar quantizer and possibly the mode of operation for each of the coding block. In other words, bit allocation tries to achieve the best visual quality without any control on the overall bit rate, while bit rate control adjusts the bit rate without any concern on the visual quality.

In order to handle the conflicting requirements of bit rate control and bit allocation in video coding, operational

rate-distortion(R-D) based framework [2-8] has attracted considerable interest in recent years. In the framework, some kind of bit rate control is applied to some relatively large coding units such as slice, frame, or even group-of-frame while R-D based bit allocation is applied to the macroblocks within the particular coding unit. The advantage of the R-D approach is that it gives an optimal solution for minimizing coding distortion under a bit-budget constraint. The R-D framework can be applied to the intra/predictive/interpolative mode selection for MPEG blocks [8] and is claimed to achieve 25% of bit saving while maintaining similar image quality. However, one drawback of the R-D approach is the heavy computational requirement because both the distortion and the bit rate need to be computed for all the possible quantization levels (or step size) in a coding unit. In MPEG1, the smallest coding unit for setting the quantization step size is a macro-block and each macro-block has thirty-one possible values of the quantization levels (*MQANT*). To compute the mean square error and bit rate for each value of *MQANT* using the brute force method requires a lot of computation (1.6×10^8 multiplications for CIF format at 10 frames/s) which is impractical. Some researchers [8] attempted to circumvent this computation problem by degenerating the macro-block level quantization into a frame level decision. This reduces much of the computational requirement but limits severely the adaptability and flexibility of the coding within a frame resulting in compromised image quality.

In this paper, we propose a novel piecewise linear approximation to the operational R-D function that is much lower in computation than the brute force method. In our approach, no a priori knowledge of R-D function is assumed. Computational saving is achieved by computing the distortion and the bit rate only for selected points using a fast iterative algorithm. The R-D function is then interpolated linearly based on the selected points. The proposed scheme is defined in the framework of MPEG-1, though the extension to the other standards is straightforward. In section 2, distortion measures are defined. In section 3, the proposed fast piecewise estimation of the distortion function for intracoded and non-intracoded quantizers are described. In section 4, the modeling and relative computation complexity of the proposed method is compared with the exhaustive methods. In section 5, simulation is done by applying the proposed R-D function model to the MPEG-1 video compression. Finally, conclusion and future works are given in section 6.

2. DISTORTION DEFINITION

In MPEG-1, there are two types of quantizers, namely the Intracoded quantizer (*I-quantizer*) and the non-intracoded quantizer (*N-quantizer*). The *DCT* coefficients of each 8 x 8 block are uniformly quantized as

$$C_{ij}^q = op\left(\frac{8C_{ij}}{Q_p q_{ij}}\right)$$

where C_{ij} is the ij^{th} *DCT* coefficient, C_{ij}^q is the ij^{th} quantized *DCT* coefficient, q_{ij} is the ij^{th} element of the quantization matrix, Q_p is an additional integer quantization factor (usually called *MQANT* for macroblock level quantization) ranging from 1 to 31, and $op()$ is *rounding()* for *I-quantizer*, and *truncation()* for *N-quantizer*. The corresponding dequantized coefficient is

$$\tilde{C}_{ij} \equiv (2C_{ij}^q + \text{sgn}(C_{ij}^q)) \frac{Q_p q_{ij}}{16} \quad \text{sgn}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } (x > 0) \\ -1 & \text{if } (x < 0) \end{cases}$$

Typically the q_{ij} is different for different i and j but the quantization matrix is unchanged within each frame. There are default quantization matrices specified in MPEG-1, which are used in this paper. The Q_p is fixed within each macroblock, but can be different for different macroblocks. Often the Q_p is adjusted to control the

local bit rate generation and to control the local distortion level.

In general, mean square error (*MSE*) between the pixel values of the original image and the reconstructed image is commonly used as the distortion measure. Frequency weighting is sometimes applied to *MSE* to account for some subjective perceptual distortion. The proposed fast piecewise linear approximation applies equally well to both *MSE* and frequency weighted *MSE*.

Due to the unitary nature of DCT, the pixel domain mean square error (*MSE*) and the DCT-domain mean square error (*DME*) should be identical.

$$DME = \sum_{i=1}^8 \sum_{j=1}^8 e_{ij}^2 = \sum_{i=1}^8 \sum_{j=1}^8 (C_{ij} - \tilde{C}_{ij})^2 = \sum_{i=1}^8 \sum_{j=1}^8 (x_{ij} - \tilde{x}_{ij})^2 = MSE \quad (1)$$

where x_{ij} , $\tilde{x}_{ij}$ are the pixels of the original and reconstructed blocks respectively, C_{ij} and $\tilde{C}_{ij}$ are the ij^{th} original and dequantized DCT coefficients, and $e_{ij} \equiv |C_{ij} - \tilde{C}_{ij}|$ is the DCT domain absolute error. In the proposed scheme, a fast iterative algorithm is used to recursively compute e_{ij} for the selected values of *MQ*UANT.

3. PIECEWISE LINEAR APPROXIMATION OF RATE DISTORTION FUNCTION

To generate the rate function, the quantity $8C_{ij}/q_{ij}$ is first computed for each ij . To obtain the rate for any Q_p , the C_{ij}^q is obtained by dividing $8C_{ij}/q_{ij}$ by Q_p followed by the rounding or truncation. The quantized DCT coefficients C_{ij}^q are zig-zag scanned and variable length encoded. The variable length coder is essentially a table lookup. For an 8x8 block, the computation of $8C_{ij}/q_{ij}$ takes 128 multiplications, which can be re-used in the distortion function generation below. The computation of C_{ij}^q takes at most 64 multiplication for each Q_p or *MQ*UANT. However, most of the high frequency C_{ij}^q are zero when Q_p is large and thus much of the multiplication can be saved. As the computation requirement of the rate function generation is much less than the distortion function generation, we will focus more on the fast distortion function generation.

On the other hand, the computation of the distortion function takes much more computation. The brute force approach to generate the *DME* for each of the 31 *MQ*UANT is to actually compute the C_{ij}^q for each *MQ*UANT as in the rate function generation, compute the $\tilde{C}_{ij}$ which involves two more multiplication for each non-zero DCT coefficient and then compute the *DME*. Much computation (64 multiplications and 127 additions) is needed to compute *DME* according to Eqn. (1) for each *MQ*UANT.

In the following, we propose to compute the *DME* only for six values of *MQ*UANT, namely, {1, 2, 4, 8, 16, 32} and then linearly interpolate the other values. Notice that 32 is not an allowable value for *MQ*UANT in MPEG-1 but we use it because the *DME* corresponding to 32 can be computed with little computation. Two slightly different algorithms are required to handle *I-quantizer* and *N-quantizer* separately.

For the *I-quantizer*:

(Initialization)

1. For $Q_p=1$, compute the following three quantities

$$M_{ij,1} = \text{int}(10(8C_{ij}/q_{ij}))$$

$$C_{ij,1}^q = \text{round}(M_{ij,1}/10)$$

$$e_{ij,1} = |C_{ij,1}^q q_{ij}/8 - C_{ij}|$$

for all i and j , where $\text{int}(\cdot)$ is truncation and $M_{ij,1}$ is an integer that keeps the first decimal place of $(8C_{ij})/q_{ij}$. The ± 1 adjustment on even reconstructed values is ignored as it has small error when the quantization step size is large.

2. Compute DME_1 as the square sum of the $e_{ij,1}$.
3. Compute $RATE_1$ from the $e_{ij,1}$ using the Huffman table.

(Recursion)

4. Assume $M_{ij,n}$, $C_{ij,n}^q$ and $e_{ij,n}$ are available corresponding to $Q_p=2^n$. Compute

$$M_{ij,n+1} = M_{ij,n}/2$$

$$C_{ij,n+1}^q = \text{round}(M_{ij,n+1}/10)$$

for each nonzero $M_{ij,n}$, for $Q_p=2^{n+1}$.

5. Compute $e_{ij,n+1}$ as follows:

$$e_{ij,n+1} = \begin{cases} e_{ij,n} & \text{for } \text{round}(8C_{ij}/2^n q_{ij}) \text{ even} \\ e_{ij,n} + 2^n q_{ij}/8 & \text{for } \text{round}(8C_{ij}/2^n q_{ij}) \text{ odd and} \\ & \text{round}(8C_{ij}/2^n q_{ij}) < 2\text{round}(8C_{ij}/2^{n+1} q_{ij}) \\ e_{ij,n} - 2^n q_{ij}/8 & \text{otherwise} \end{cases}$$

6. Compute DME_n for $Q_p=2^n$. Compute $RATE_n$.
7. If n is less than 6, increment n and goto step 4. Otherwise, goto step 8.
8. Obtain the DME and $RATE$ for the other Q_p using linear interpolation.

For the N -quantizer :

(Initialization)

1. For $Q_p=1$, compute the following two quantities for all the i and j .

$$C_{ij,1}^q = M_{ij,1} = \text{int}(8C_{ij}/q_{ij})$$

$$e_{ij,1} = (C_{ij} - (2C_{ij,1}^q + 1) - q_{ij}/16)$$

2. Compute DME_I as the square sum of the $e_{ij,1}$.
3. Compute $RATE_I$ from the $e_{ij,1}$ using the Huffman table.

(Recursion)

4. Assume $M_{ij,n}$, $C_{ij,n}^q$ and $e_{ij,n}$ are available corresponding to $Q_p=2^n$ and $q_{ij}=16$. Compute

for each nonzero $M_{ij,n}$, for $Q_p=2^{n+1}$.

5. Compute $e_{ij,n+1}$ as follows:

$$e_{ij,n+1} = \begin{cases} e_{ij,n} & \text{for } C_{ij,n}^q \text{ even and } C_{ij,n}^q = 0 \\ e_{ij,n} - 2^n \text{sgn}(C_{ij}) & \text{for } C_{ij,n}^q \text{ even and } C_{ij,n}^q \neq 0 \\ e_{ij,n} + 2^n \text{sgn}(C_{ij}) & \text{for } C_{ij,n}^q \text{ odd and } \text{abs}(C_{ij,n}^q) \geq 2 \\ e_{ij,n} + (3(2^n) - 1) \text{sgn}(C_{ij}) & \text{for } C_{ij,n}^q \text{ odd and } \text{abs}(C_{ij,n}^q) < 2 \end{cases}$$

6. Compute DME_n for $Q_p=2^n$. Compute $RATE_n$.
7. If n is less than 6, increment n and goto step 4. Otherwise, goto step 8.
8. Obtain the DME and $RATE$ for the other Q_p using linear interpolation.

4. COMPUTATIONAL COMPLEXITY

For an intracoded 8x8 block, it takes 128 multiplications and 64 roundings to perform the uniform quantization for one Q_p , and another 192 multiplications, 127 additions and 64 absolute conversions to obtain the DME . Altogether, it takes 320 multiplications, 127 additions, 64 roundings and 64 absolute conversions for each Q_p . To obtain the MSE - Vs - Q_p curve for the same 8x8 block using brute force computation of DME for all Q_p values, it takes 9920 multiplications, 3937 additions, and 1984 roundings. For a CIF (352x288) sized video at 10 frames per second, the total computation required by the brute force method is 1.6×10^8 multiplications,

6.2×10^7 additions, and 3.1×10^7 roundings, which makes the method too expensive to implement.

To obtain the piecewise linear approximation of the MSE -Vs- Q_p curve using the interpolation of DME for $Q_p=1, 2, 4, 8, 16, 32$, it takes 576 multiplications, 511 additions, 384 roundings, 63 shifting. This translated to a computation reduction factor of 17, compared with brute force computation. Note that the recursive procedure applies only to nonzero $C_{ij,n}^q$ of which there are progressively fewer as Q_p increases, especially for high frequency. Typically there are very few left for $Q_p=8, 16$ and 32. For the sake of estimating computational complexity, we have assumed that the number of nonzero $M_{ij,n}$ is reduced by a factor of two in each iteration. If a CIF (352x288) sized video at 10 frames per second is completely intra-coded, the total computation required by the proposed fast rate-distortion function generation is 9.1×10^6 multiplication, 8.1×10^6 addition, 6.1×10^6 rounding and 1.0×10^6 shifting. This is still a lot of operations, but is at least close to practical to implement.

Similarly, for a non-intracoded 8x8 block, the brute force approach requires 9920 multiplications, 9856 additions, and 3968 if-then clause to compute the DME . The proposed fast algorithm requires 388 multiplications, 1284 additions, 1152 if-then clause and 160 shifting. This corresponds to a computation reduction factor of about 20. Again, we have assumed that the number of nonzero $M_{ij,n}$ is reduced by a factor of two in each iteration. If a CIF sized video at 10 frames per second is completely non-intracoded, the total computation of the proposed algorithm is 6.1×10^6 multiplications, 2.0×10^7 additions, 1.8×10^7 if-then clause and 2.5×10^6 shifting which is close to practical to implement.

The proposed rate-distortion approximation and the brute force computation of the R-D curve are performed on a SUN SPARC 10 workstation using the "Football" and "Tennis" sequence which are in CIF format. The results are tabulated as follows :

Table 1: CPU Time on Sun SPARC 10

	Football	Tennis
I-frame (Proposed)	3.0 s	2.5 s
I-frame (Brute force)	52 s	51.5 s
P-frame (Proposed)	2.0 s	2.0 s
P-frame (Brute force)	42 s	42 s

Table 2: Computation Complexity

	Football	Tennis
I-frame (Proposed)	1.00	1.00
I-frame (Brute force)	15.43	16.27
P-frame (Proposed)	1.00	1.00
P-frame (Brute force)	16.28	17.10

Table 3: Modeling Error=|fitted value-actual value|/actual value

	Football	Tennis
I-frame (MSE)	3.68%	3.36%
I-frame (Rate)	7.00%	12.10%
P-frame (MSE)	4.70%	4.72%
P-frame (Rate)	28.60%	35.70%

The actual CPU time for the computation of the rate-distortion function for one frame is given in Table 1. In the same simulation, the number of multiplications were counted and shown in Table 2. Here the amount of multiplication for the proposed algorithm is expressed as unity to shown how much slower the brute force algorithm is. It can be observed that the proposed algorithm achieves an average reduction factor of 15 to 17, which verifies our computation complexity analysis. Our assumption that the number of nonzero $M_{ij,n}$ is reduced by a factor of two in each iteration appears to be at least reasonable, though it would be always image dependent.

Two typical graphs showing the relationship between the piecewise model and the full exhaustive computation of R-D function are shown in Figure 1 where the circles refer to the true value calculated by brute force while the line is fitted using the proposed piecewise approach. In general, the proposed linear approximation can fit the true curve very closely. The average modeling error defined as

$$\text{modeling error} = |\text{fitted value} - \text{actual value}| / \text{actual value}$$

is shown in Table 3. The proposed piecewise linear approximation of the distortion function gives a modeling error of 3% to 5%, which is quite reasonable. The error is observed to be distributed quite uniformly among the 31 Q_p coefficients. However, the piecewise linear approximation does not seem to be appropriate for the rate function. The modeling error appears to be considerably lower in I-frames. This is probably because P-frames are residue frames with low energy content, which means the blocks are coded with much lower bitrate. The modeling error tends to be large when the actual value is small. However, the large modeling error can be reduced to zero by simply computing the rate for all the possible $MQUANT$. As discussed before, the computa-

Figure 1 Two examples of rate-distortion curve generated by brute force and proposed algorithm

5. SIMULATION ON VIDEO CODING

In this section, we show by simulation the effectiveness of the operational rate distortion method and compare

the performance change as the brute force rate-distortion function generated is replaced by the proposed fast piecewise linear approximation. The operational rate distortion method we used in the simulation is as follows.

Let Q_i be the quantization step size assigned for macroblock i . Let $D(Q_i)$ and $R(Q_i)$ be the distortion level and bit rate of macroblock i respectively. The constrained bit allocation problem is :

$$\min(\sum_i D_i(Q_i)) \text{ for all macroblock } i \text{ within a slice of a video frame such that } \sum_i R_i(Q_i) \leq R_{budget}$$

The above constrained problem can be converted to an unconstrained problem as follows :

$J(\lambda) = \min(\sum_i (D_i(Q_i) + \lambda R_i(Q_i)))$ for all macroblock i within a slice, where $\lambda \geq 0$ is the Lagrange Multiplier and $J(\lambda)$ is the cost function.

For a given $\lambda = |(\Delta D)/(\Delta R)|$, it corresponds to the slope of a point on the R-D curve. If λ is changed from 0 to ∞ , it will trace through all the operating points in the R-D curve. Figure 2 shows a R-D curve for a macroblock i with a tangent at $(D_i(Q_i), R_i(Q_i))$, the coordinate that corresponds to $\min(\sum_i (D_i(Q_i) + \lambda R_i(Q_i)))$.

The slope of the tangent is $-\lambda$ and the y-intercept is $J(\lambda)$ for that macroblock i .

Figure 2 Rate distortion curve

If the encoding of macroblocks are independent of each other, the minimization can be done independently for each i . As a result,

$$J(\lambda) = \sum_i (\min(D_i(Q_i) + \lambda R_i(Q_i))) \quad (2)$$

Graphically, the meaning of Eqn. (2) can be illustrated using Figure 3. A matrix of dots are shown where each column corresponds to the macroblock i increasing from left to right and each row represents the setting of Q_i increasing from top to bottom. Each dot in the figure is the value of $D_i(Q_i) + \lambda R_i(Q_i)$. For each macroblock i , the minimization in Eqn. (2) is to select the Q_i which minimizes $D_i(Q_i) + \lambda R_i(Q_i)$. This Q_i is defined as

$Q_{i,min}$ for a particular λ . Graphically, it can be represented as a marked dot in the column i of Figure 3 where the row number of the marked dot is the selected $Q_{i,min}$. As the minimization is done independent of other macroblocks, a set of $Q_{i,min}$ for different macroblocks is obtained. In Figure 3, the set is shown using pointed arrow as a path from the first to the last macroblock within the set.

Figure 3 Graphical representation of Trellis path

Having determined $J(\lambda)$, our goal is to select λ such that $R_{budget} = \sum_i R_i(Q_{i,min})$. With the selected λ , the set of $Q_{i,min}$ is obtained as the quantization step size settings for the macroblocks within the set. Usually we cannot select a λ which achieve the R_{budget} exactly but we can find one which is the closest to but less than it.

Methods to search for the optimal λ have been investigated in [4,5]. Iterative methods such as bisection search and fast convex search, can be used. The bisection approach is used in our simulation and is outlined here.

1. $D(Q_i), R(Q_i)$ are evaluated for all possible Q_i values (1 to 31) of all the macroblocks within a picture frame.
2. Let $\lambda_l = 0, \lambda_h = \infty$
3. Evaluate $J(\lambda_l)$ and obtain $Q_{il,min}$ which is the set of optimal Q_i and $R_l = \sum_i R_i(Q_{i,min})$ for λ_l
4. Evaluate $J(\lambda_h)$ and obtain $Q_{ih,min}$ which is the set of optimal Q_i and $R_h = \sum_i R_i(Q_{i,min})$ for λ_h
5. If Target Rate is greater than or equal to R_l then select $Q_{il,min}$ as the solution; else goto step 6.
6. If Target Rate is lower than or equal to R_h then select the $Q_{ih,min}$ as the solution; else goto step 7.
7. The following pseudo program is executed to obtain the solution.

```

While (Not Finished) do
    Evaluate  $\lambda_t$  from  $Q_{il,min}$  and  $Q_{ih,min}$ 
    Evaluate  $J(\lambda_t)$  and obtain  $Q_{it,min}$  and  $R_t = \sum_i R_i(Q_{i,min})$  for  $\lambda_t$ 
    If ( $R_t > \text{Target Rate}$ ) then
         $\lambda_l = \lambda_t$ 
         $Q_{il,min} = Q_{it,min}$ 
         $R_l = R_t$ 
    Else If ( $R_t < \text{Target Rate}$ ) then
         $\lambda_h = \lambda_t$ 
         $Q_{ih,min} = Q_{it,min}$ 
         $R_h = R_t$ 
    Else If ( $R_t = \text{Target Rate}$ ) then
         $Q_{it,min}$  is the solution
        Finished = True
    End if
End While

```

Using the bisection approach, the optimal quantization step size for a set of macroblocks can be obtained. In the simulation, the number of iteration is found to be very small usually. The computational requirement of the bisection method is found to be quite small as compared with the generation of the rate distortion curve.

In the simulation, 208 frames of the "Football" sequence in CIF format are encoded using the Belcore MPEG-1 software simulator[9]. The bitrate is set at 1.15Mbps, and the group of pictures is 15. The simulation is first performed with the built-in rate control algorithm but without any operational rate-distortion based bit allocation. The bit allocation to each of the 208 frames are recorded. The simulation is then repeated using the operational rate-distortion based bit allocation with the same bit allocation to each of the 208 frames. Brute force computation of the rate-distortion function is used. Another simulation is performed with the same set up except that the proposed piecewise linear approximation of the rate-distortion function is used instead. Here the rate distortion optimization is done on a frame basis where the bits allocated to each frame is the same as that of the original simulator.

The PSNR of the three cases are plotted against the frame number in the Figure 4. The average PSNR of the original Belcore simulator is 31.63dB. By incorporating the operational R-D based bit allocation to each frame, the PSNR becomes 32.78dB, a significant increase of 1.15dB. The PSNR is higher than that of the original simulator in essentially all the frames. This suggests that a significant increase in visual quality is possible by incorporating the R-D based bit allocation. However, the huge computation requirement makes it too expensive to implement. By using the proposed fast piecewise linear approximation of the R-D function, the average PSNR is reduced to 32.27dB. The price of the significant computation reduction by a factor of about 17 is a smaller gain in PSNR (0.64dB compared with a possible 1.15dB). Further work is needed to improve the accuracy of the proposed algorithm without increasing the computational complexity.

Figure 4 PSNR of football sequence

6. CONCLUSION

In this paper, we propose a novel efficient way to estimate the mean square quantization distortion function as a function of the quantization step size MQ_{ANT} (or Q_p) in the MPEG-1 encoder. We use piecewise linear modeling to approximate MSE . To reduce the computation requirement, we propose to compute DAE only for six Q_p values, $Q_p=1, 2, 4, 8, 16, 32$, and square the entry terms to obtain the MSE and other values are interpolated by piecewise linear interpolation. The recursive algorithm is different for intracoded and intercoded blocks. A computation reduction factor of about 17 is possible. The simulator with operational Rate-Distortion bit allocation is found to have an advantage of 1.15dB in terms of PSNR over the original simulator. The proposed fast algorithm is found to provide a good tradeoff option. The proposed fast algorithm can reduce the computation of the brute force method significantly at the price of considerable decrease in PSNR.

7. ACKNOWLEDGEMENT

This work was supported in part by an RGC grant and an HKTIIT grant.

8. REFERENCE

1. L. Wang, "Rate Control for MPEG Video Coding", *Proc. of SPIE Visual Communication and Image Processing*, Vol-1, pp.53-64, May 95.
2. T. Wiegand, et. al., "Efficient Mode Selection for Block-based Motion Compensated Video Coding", *Proc. of IEEE Int. Conf. on Image Processing*, Vol-2, pp. 559-562, Oct. 95.

3. J.B. Cheng, H.M. Hang, "Adaptive Piecewise Linear Bits Estimation Model for MPEG Based Video Coding", *Proc. of IEEE Int. Conf. on Image Processing*, Vol-2, pp. 551-554, Oct. 95.
4. K. Ramchandran, A. Ortega, M. Vetterli, "Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Coders", *IEEE Transactions on Image Processing*, Vol. 3, No.5, pp. 533-545, Sept 94.
5. A. Ortega, K. Ramchandran, M. Vetterli, "Optimal Trellis-Based Buffered Compression and Fast Approximations", *IEEE Transactions on Image Processing*, Vol.3, No.1, pp. 26-40, Jan 94.
6. A. Ortega, "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to MPEG video coders", *Proc. of SPIE*, Vol. 2419, pp. 129-38, Feb. 95.
7. E.D. Frimout, J. Biemond, R.L. Lagendijk, "Forward rate control for MPEG recording", *Proc. of SPIE*, Vol. 2094, Pt. 1, pp.184-194, Nov. 93.
8. W. Kwok, H. Sun, J. Ju, "Obtaining an Upper Bound in MPEG Coding Performance from Jointly Optimizing Coding Mode Decisions and Rate Control", *Proc. of SPIE*, Vol. 2501, Pt. 1, pp. 2-10, May 95.
9. Bellcore MPEG-1 software simulator, 90.