Performance and Computational Complexity Optimization in Configurable Hybrid Video Coding System

David Nyeongkyu Kwon, *Senior Member, IEEE*, Peter F. Driessen, *Senior Member, IEEE*, Andrea Basso, and Pan Agathoklis, *Senior Member, IEEE*

***Abstract*—In this paper, a configurable coding scheme is proposed and analyzed with respect to computational complexity and distortion (C-D). The major coding modules are analyzed in terms of computational C-D in the H.263 video coding framework. Based on the analyzed data, operational C-D curves are obtained through an exhaustive search, and the Lagrangian multiplier method.**

The proposed scheme satisfies the given computational constraint independently of the changing properties of the input video sequence. A technique to adaptively control the optimal encoding mode is also proposed. The performance of the proposed technique is compared with a fixed scheme where parameters are determined by off-line processing. Experimental results demonstrate that the adaptive approach leads to computation reductions of up to 19%, which are obtained with test video sequences and compared to the fixed, while the peak signal-to-noise ratio degradations of the reconstructed video are less than 0.05 dB.

***Index Terms*—Complexity distortion optimization, dynamic programming (DP), hybrid video coding, Lagrangian relaxation, optimal resource allocation.**

I. Introduction

MULTIMEDIA communications involving audio, video and data has been an interesting topic because of the many possible applications. Recently, hardware platforms for hand-held devices such as PDAs have improved dramatically, which has created a special interest in implementing videos in portable devices. However, video-coding algorithms are still much too complex for implementation in hand-held devices, which are powered by batteries with a limited storage capacity. Therefore, computationally configurable video coding schemes would be beneficial for such constrained environments.

The question is how to achieve optimal computing resource allocation among encoding modules for given computational constraints, so that the system can make the best use of limited computing resources to maximize its coding performance in terms of its video quality. Work in the area of optimal video coding is reviewed in [1] and [2]. One of the common approaches is to optimize the bit allocation by taking into account the resulting rate and distortion. Although this is a good approach to deal with bandwidth limitations, this may not give good performance where the computational complexity is the main limitation.

The rate distortion optimization problem in a video coding framework is addressed in [3] and [4], where motion estimation (ME), mode decision, and quantization are considered either separately or jointly for the best tradeoff. Although complexity is addressed in conjunction with rate and distortion, only the discrete cosine transform (DCT) and inverse DCT (IDCT) modules of the video coding system are considered [5], [6].

In this paper, the performance of a configurable video system is analyzed with respect to computational complexity and distortion (C-D). The system consists of three coding modules, each having a control parameter (such as window size in ME) controlling the computational complexity and the quality of the reconstructed video sequence. The approach considered here is different from the one in [7], where an iterative method is used to find the optimal control variables. More specifically the method in [7] measures the system complexity in terms of averaged frames per second, while the one proposed in [5], [6] gives the predetermined complexity of the coding system regardless of the varying input contents and sequence. [35] introduces a baseline framework of the proposed concept and presents interim results. Based on the previous work, we here extend it to an adaptive scheme whereby more accurate control parameters are found particularly with active sequences.

This approach could be reasonably accurate enough to estimate the system complexity as far as major coding modules are taken into account in the system configuration. The C-D data is obtained by analyzing the operations required for each module, and by evaluating the distortion in the reconstructed sequence for the possible control parameter values.

This paper is organized as follows. In Section II, a general formulation of the optimization problem is presented. In Section III, the computational C-D of major coding modules are analyzed. An operational C-D curve is obtained using the analyzed data from test video sequences, and an adaptive control scheme is introduced in Section IV. Finally, its implications for the performance of the coder are discussed, and concluding remarks given, in Section V.

Manuscript received March 13, 2003; revised April 16, 2004 and December 22, 2004. This work was supported in part by AVT Audio Visual Telecommunications Company, Victoria, BC, Canada. This paper was recommended by Associate Editor H. Sun.

D. N. Kwon, P. Driessen, and P. Argathoklis are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: nkwon@ece.uvic.ca; peter@ece.uvic.ca; pan@ece.uvic.ca).

A. Basso is with NMS Communications, Red Bank, NJ 07701 USA (e-mail: andrea_basso@nmss.com).

Digital Object Identifier 10.1109/TCSVT.2005.858615

II. General Problem Formulation

Consider a video coding system that is decomposed in N modules $M_1, \ldots, M_N$. Each module $M_i, i = 1, \ldots, N$, is assigned a control variable s_i, which determines both the computational complexity required for coding and the distortion of the reconstructed video sequence. Each control variable s_i can take k_i distinct values from the set $S_i = \{s_{ij} | j = 1, \ldots, k_i\}$ for $i = 1, \ldots, N$. With these definitions, it is now possible to express the computational complexity $C(s_1, \ldots, s_N)$ for the video coding system as

$$C(s_1, \ldots, s_N) = \sum_{i=1}^{N} c_i(s_i) \tag{1}$$

where $c_i(s_i)$ is the computational complexity for each coding module $M_i, i = 1, \ldots, N$. The complexity for each coding module depends on the control variable for this module s_i.

The distortion between the original and the reconstructed video sequence can be represented as $D(s_1, \ldots, s_N)$. Each coding module $M_i, i = 1, \ldots, N$, contributes to $D(s_1, \ldots, s_N)$ even though the individual contributions are not additive. The distortion depends again on the control variable s_i for each module M_i.

The problem considered here is finding the control variable values for the N coding modules, which would lead to minimal distortion of the reconstructed video sequence for a given limited computational complexity. This can be formulated as follows:

$$\min_{[s_1,\ldots,s_N] \in S_1 \times,\ldots,\times S_N} D(s_1, \ldots, s_N) \tag{2}$$

subject to $C(s_1, \ldots, s_n) \leq C_{\max}$.

This is a constrained optimization problem where the optimization variable $s_1, \ldots, s_N$ can take distinct values. A known approach [26]–[31] to solve this constrained optimization problem is to consider the following unconstrained optimization problem

$$\min_{[s_1,\ldots,s_N] \in S_1 \times,\ldots,\times S_N} D(s_1, \ldots, s_N) + \lambda C(s_1, \ldots, s_N) \tag{3}$$

where the Lagrangian multiplier λ is a nonnegative number. It is well known in operational research that the Lagrangian relaxation method will not necessarily give the optimal solution, since the Lagrangian multiplier λ can reach only the operating points belonging to the convex hull in the operational complexity-distortion curve. When λ sweeps from 0 to infinity, the solution to problem (3) traces out the convex hull of the complexity distortion curve.

The Lagrangian multiplier λ allows a tradeoff between C-D. When $\lambda = 0$, minimizing the Lagrangian cost function is equivalent to minimizing the distortion. Conversely, when λ becomes large, the minimization of the Lagrangian cost function is equivalent to minimizing the complexity. Many fast algorithms have been developed by several authors [32]–[34] to find the optimal λ. Hence, assuming an optimal Lagrangian multiplier for the given computational constraint is given through either a fast or an exhaustive search of the Lagrangian multiplier, the problem now is to find the optimal solution to the unconstrained problem of (3).

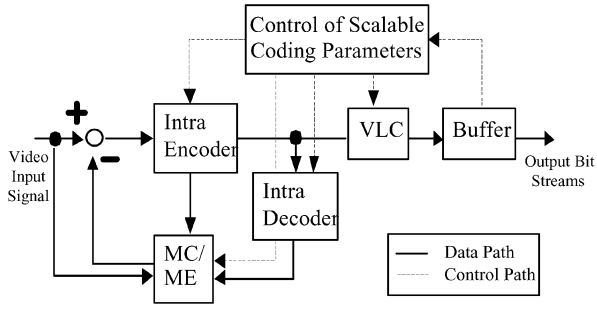


Fig. 1. Configurable coding scheme with scalable coding parameters.

In this analysis, a configurable video coding scheme like the one outlined in Fig. 1 is considered. For our analysis it is assumed that the system consists of three major coding modules with corresponding control variables:

- M_1 ME module where the control variable s_1 can take values from the set $S_1 = \{0, \ldots, 3\}$ corresponding to variable search range, $p \in \{3, 5, 7, 9\}$, respectively;
- M_2 integer or fractional (I/F) pixel accuracy in ME, where the control variable can take the values $s_2 = 0$ (integer) or $s_2 = 1$ (fractional) pixel accuracy;
- M_3 DCT where the control variable s_3 can take values from the set $S_3 = \{0, \ldots, 3\}$ corresponding to different DCT coefficient pruning options $w \in \{2, 4, 6, 8\}$, respectively.

III. Complexity and Distortion Analysis

In this section, the computational complexity of each of these coding modules is evaluated. Our complexity computation considers all processor instructions, including multiplications and additions with the same weighting factor as one instruction, as in [12]. Since we are interested in the relative complexity and accuracy, the computational complexity for only one frame is computed.

A. ME Module

There are many block-matching fast search algorithms, such as TSS [10], 2-D LOG [9], DS [11], [12], Conjugate Directional Search (CDS) [42], and so on, which have been developed to reduce the computational complexity of a full exhaustive search algorithm. TSS is one of the fast search algorithms, reducing computational complexity to $8 \log p$, where p is the search range parameter. The size of the initial step, and the next, is calculated by dividing the search range parameter p by 2 in each. The number of search points is eight in each step, except in the initial one, which needs one more point in the zero vector location. Note that the computational complexity of TSS given in the number of search points is constant, not changing with the varying contents in the video sequence. In TSS, the search points are pre-defined for all macroblocks, as shown in Fig. 2. Other algorithms, such as DS and CDS, search for the motion vector of the macroblock starting from the zero vector location until the best motion vector are found that meet the given cost measure, the locations and the total number of search points changes for each macroblock.

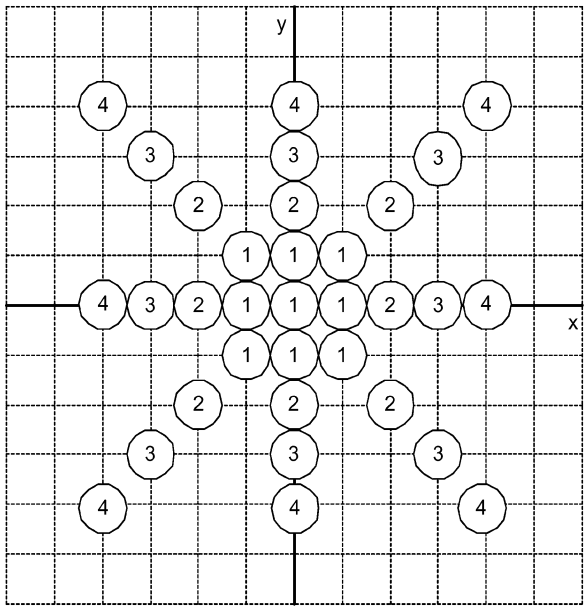


Fig. 2. Search points according to the different search window in the three-step search.

This deterministic property of TSS can be used in implementing a configurable coding system with a hard-control feature. Therefore, this search range parameter is chosen as a control parameter in a tradeoff between complexity and accuracy. Fig. 2 shows the number of search points with regard to the search range, where zero vector MV(0, 0) is assumed as the real vector giving the minimum cost function. The numbers 1, 2, 3, and 4 in the figure, which mean the window size of the motion vector search, correspond to 3×3, 5×5, 7×7, and 9×9, respectively.

The complexity analysis here is based on a frame size of 176×144 QCIF format, a block size of 16×16 and the use of the Mean Absolute Difference (MAD) as the matching criterion. The MAD calculation can be represented as below.

$$\mathrm{MAD}(dx, dy) = \frac{1}{N^2} \sum_{i=\frac{-N}{2}}^{N/2} \sum_{j=\frac{-N}{2}}^{N/2} \times |F(i,j) - G(i+dx, j+dy)| \quad (4)$$

where $F(i,j)$ is the $N \times N$ macroblock being compressed; $G(i,j)$ is the reference $N \times N$ macroblock, and dx and dy are the search location motion vectors; N is the macroblock size. The evaluation of each MAD cost function requires 2×256 load operations, 256 subtraction operations, one division operation, one store operation and one data compare operation, for a total $2 \times 256 + 256 + 1 + 1 + 1 = 1035$ operations [12]. The overall computational complexities according to different search ranges are analyzed in Table I.

B. I/F Module

The accuracy of the motion vectors obtained can be improved using half pixel accuracy [10]; that is, by using eight surrounding half-pixels from the integer pixel location. First, computing operations for bilinear interpolation per macro block are 324 data loads, 162 additions, 162 divisions, 486 data accumulations, and 162 data divisions, for a total of 1296 operations. Therefore, for the QCIF format and block size of 16×16, the total number of operations for a half-pel search can be evaluated as follows:

(Total number of operations per MAD cost function
Number of search locationssurrounding integer motion vector
+ Bilinear interpolation per integer motion vector)
(Number of macro blocks) $= (779 \times 8 + 1296) \times 99$
$= 745,272$ (5)

TABLE I
COMPUTATIONAL COMPLEXITY AS A FUNCTION OF THE SEARCH WINDOW SIZE FOR THE ME SEARCH USED

Search Windows size, S_1	S_1 =0	S_1 =1	S_1 =2	S_1 =3
	[-3,3]	[-5,5]	[-7,7]	[-9,9]
Search Points	9	17	25	33
Computations	694,089	1,311,057	1,928,025	2,544,993

C. DCT Module

DCT has been used for most image and video coding standards because its energy compaction performance is close to that of Karhunen–Loeve Transform (KLT), known as the optimum image transform in terms of energy compaction, sequence entropy and de-correlation. Most of the energy is compacted into the top left corner, so that the least number of elements are required for its representation. The basic computation of the DCT-based video and image compression system is the transformation of an 8×8 image block from the spatial domain to the DCT transform domain. The two-dimensional (2-D) 8×8 transformation is expressed by (6) [14]

$$y(k,l) = \frac{c(k)c(l)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} x(i,j) \cos \frac{(2i+1)k\pi}{16} \cdot \cos \frac{(2j+1)l\pi}{16}, \qquad k,l = 0, \ldots, 7 \quad (6)$$

where $c(k) = 1/\sqrt{2}$ for $k = 0$ and $c(k) = 1$ otherwise. The 2-D DCT transform can be decomposed into two one-dimensional (1-D) 8-point transforms, as (6) can be modified as

$$y(k,l) = \frac{c(k)}{2} \sum_{i=0}^{7} \left[\frac{c(k)}{2} \sum_{j=0}^{7} x(i,j) \cos \frac{(2j+1)k\pi}{16} \right] \cdot \cos \frac{(2i+1)l\pi}{16}, \qquad k,l = 0, \ldots, 7 \quad (7)$$

where $[\cdot]$ denotes the 1-D DCT of the rows of input $x(i,j)$.

Regarding computational complexity, the 2-D DCT computation of the (6) requires 4096 multiplications and additions. However, using the row-column decomposition approach of (7), it

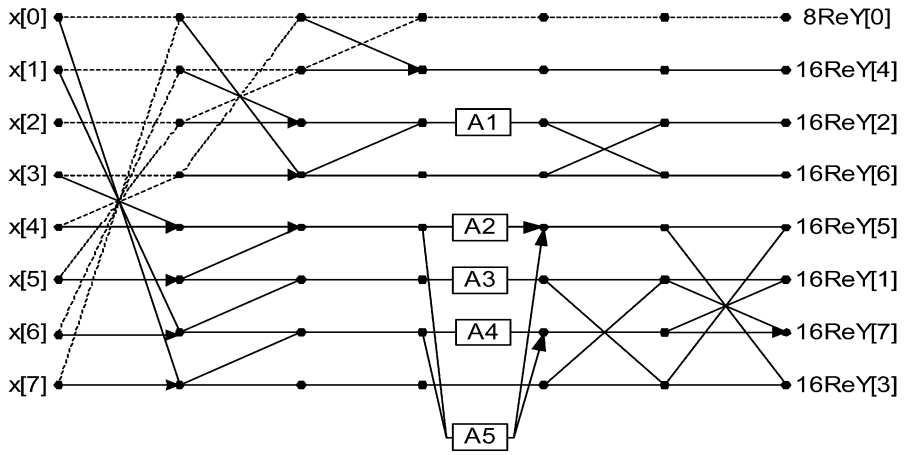


Fig. 3. AAN forward DCT flow chart where DCT pruning for y(0) coefficient is represented by the dotted-line.

can be reduced to 1024 multiplications and additions, four times less than that of (6). Although the separability property of DCT has reduced the computational complexity, these numbers are still prohibitive for real-time application. Until now, many fast DCT computation algorithms [20]–[22] have been developed utilizing transform matrix factorization as well as the previously developed fast discrete fourier transform (FFT). However, since the quantizer follows the DCT computation unit in most image and video coding systems, the DCT computational complexity can be further reduced. All of the multiplications occurring in the last stage of transform can be absorbed into the following quantizer unit. In other words, this computation yields the scaled version of the real DCT output. The computational complexities of the most commonly used fast DCT algorithms can be analyzed in the scaled-DCT approach [22]. The AAN scheme [33], adopted for the implementation of DCT pruning in this section, is the fastest implementation among the scaled 1-D DCT algorithms. It adopts the small and fast FFT algorithm developed by Winograd requiring only five multiplications and 29 additions, and is expressed as follows:

$$y(k) = \frac{2c(k)ReY(k)}{\cos\frac{n\pi}{16}} \tag{8}$$

where $c(k) = 1/\sqrt{2}$ for $k = 0$ and $c(k) = 1$ otherwise, and $\mathrm{Re}Y(k)$ are the real part of the 16-point DFT, whose inputs are double sized, with inputs $x(k)$, $k = 0, \ldots, 7$. Its flow chart for forward DCT calculation is shown in Fig. 3. Note that for real DCT data, outputs of the flow graph should be multiplied by constants in the (8). However, these multiplications, can be absorbed into the quantization process, giving overall computation reduction since DCT outputs are quantized for compression in most video and image coding systems.

One property of the DCT transform is efficient energy compaction, and the human visual system (HVS) is no more sensitive to high frequency components than the low frequency ones. These facts can be used to make computation-intensive DCT transform scaleable and controllable in its computational complexity. Some of the DCT coefficients can be pruned, since they do not need to be calculated at all. The DCT pruning reduces the computational complexity of the DCT transform, since it has an efficient energy compaction property and the most important information is kept in the low frequency coefficient. The dotted line in Fig. 3 shows required computations when DCT pruning is applied to the $y(0)$ transform coefficient, where a total of seven additions are needed. Pruning DCT transform is studied in [23] and [24].

A transform [23] derives an analytical form of computational complexity, where DCT pruning is applied to a fast 1-D DCT algorithm [25] with 12 multiplications and 29 additions. However, in this paper, AAN DCT is adopted in the computational complexity analysis of DCT pruning, since it is the best among the known 1-D DCT algorithms.

In [14], algorithmic complexity of the 2-D DCT algorithm is analyzed using row-column decompositions, which performs 1-D DCT two times for each of the rows and columns of 8×8 input data. A similar complexity measure can be applied to the AAN algorithm [22]. Table II shows the number of operations required to compute the DCT coefficients for each 8×8 block, and a frame of QCIF format when different pruning is used. In the table, 1-D and 8×8 mean 1-D 8-point and 2 D 8×8 DCT, respectively. It estimates the number of multiplications and additions as well as the total sums, with the assumption that the same weighting factor is given to both multiplication and addition. In Fig. 3, 1-D 8-point DCT requires eight data loads, five DCT coefficients, eight data stores, five multiplications, and twenty-nine additions, for a total of 55 operations. Therefore, in the 8×8 2-D block, the total number of operations becomes $2 \times 8 \times 55 = 880$ operations. It also shows how much DCT pruning performs the relative reduction of computation compared to the 8×8 full DCT.

The DCT pruning basically discards high frequency components in the transform domain, although it incurs image quality degradation. Fig. 4 shows reconstructed video frames after the DCT pruning operation. More coefficients are pruned, and more quality degradation occurs in the reconstructed frames. It is interesting to note that applying DCT pruning with a 4×4 window or an 8×8 full DCT makes little difference in terms of subjective quality, although there is a difference in the objective performance of about 1.1 dB peak signal-to-noise ratio (PSNR). This can be explained by the fact that the DCT has a property of high efficient energy compaction, and most energy is concentrated in the upper left corner. Accordingly,

TABLE II
COMPUTATION COMPLEXITY AS A FUNCTION OF PRUNING FOR THE DCT MODULE

Complexity, S_3		2x2 Pruning S_3=0			4x4 Pruning S_3=1			6x6 Pruning S_3=2			Full DCT S_3=3		
		M	A	T	M	A	T	M	A	T	M	A	T
AAN	1D	3	18	21	5	23	28	5	27	32	5	29	34
	8x8	400			588			742			880		
	Frame	158400(0.45)			232848(0.67)			293832(0.84)			348480(1.00)		

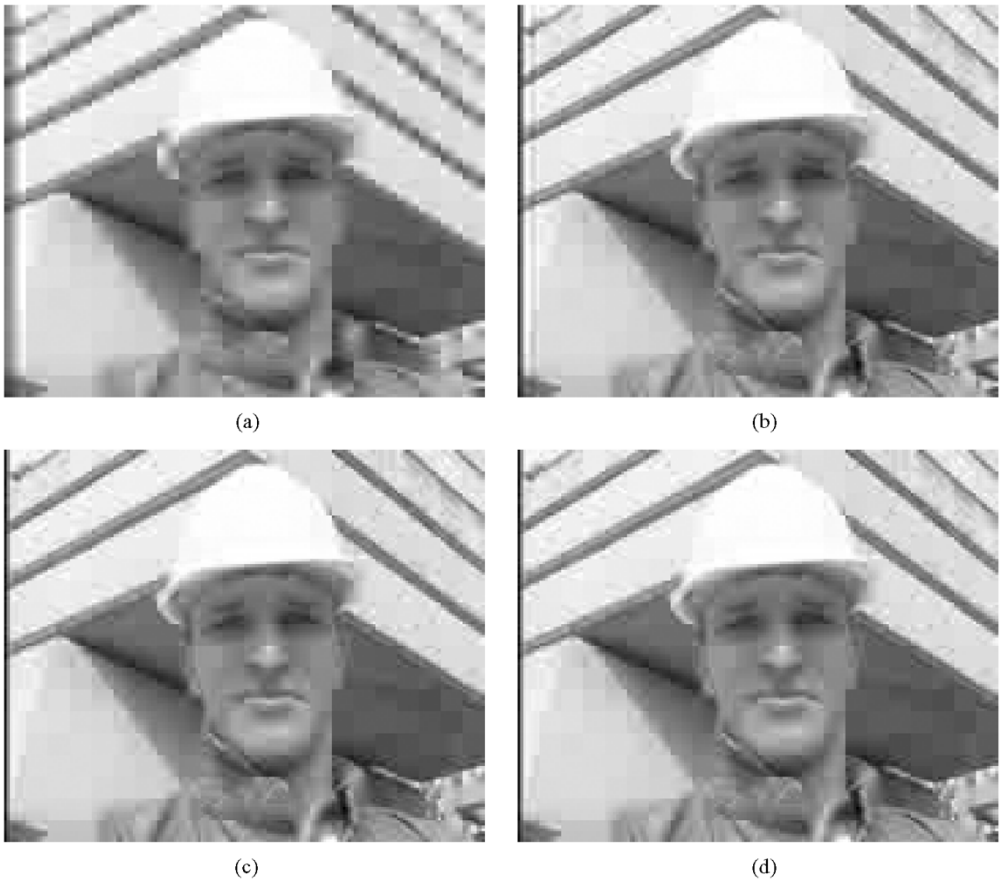


Fig. 4. Reconstructed video frames with DCT coefficient pruning (QP = 13, Intra I-frame, and H.263). (a) 2×2(25.660 dB). (b) 4×4(30.650 dB). (c) 6×6(31.739 dB). (d) 8×8 full DCT(31.740 dB).

the computational complexity of DCT can be traded off with the reconstructed image quality using, the DCT pruning.

The overall computational complexity $C(s_1, \ldots, s_N)$ can be calculated from the (1) and the above discussion, while the overall distortion $D(s_1, \ldots, s_N)$ can be estimated by exhaustive simulation for all possible operation modes of control variables, and averaged over a number of sequences and a number of frames for each sequence. In the given system, there are total 32 modes consisting of combinations of the three control variables s_1, s_2, and s_3, corresponding to ME, I/H, and DCT, respectively. Table III shows the overall computation and distortion data for all 32 operating modes. Computational complexities are represented in a total number of reduced instruction set computer (RISC)-like instructions per frame, while distortions are measured in the PSNR as follows:

$$\mathrm{PSNR} = 10 \log_{10}(255^2/\mathrm{MSD}) \tag{9}$$

$$\mathrm{MSD} = \frac{1}{N} \sum_{i=1}^{N} (O_i - R_i)^2 \tag{10}$$

TABLE III
AVERAGE PSNR DATA AND COMPUTATIONAL COMPLEXITY OF ALL OPERATION MODES WHERE FIVE VIDEO SEQUENCES WERE APPLIED AND THEIR RESULTS WERE AVERAGED

Operation Mode			Average PSNR (dB), $D(s_1,..,s_N)$	Overall Computations (1.0e+6, %), $C(s_1,..,s_N)$
ME, s_1	I/H, s_2	DCT, s_3		
0	0	0	28.13	0.734(21.72)
		1	30.53	0.752(22.27)
		2	31.21	0.768(22.73)
		3	31.48	0.781(23.13)
	1	0	29.06	1.479(43.79)
		1	31.56	1.498(44.34)
		2	32.29	1.513(44.79)
		3	32.54	1.526(45.20)
1	0	0	28.21	1.351(39.99)
		1	30.55	1.369(40.54)
		2	31.26	1.385(40.99)
		3	31.54	1.398(41.40)
	1	0	29.11	2.096(62.06)
		1	31.59	2.115(62.61)
		2	32.32	2.130(63.06)
		3	32.59	2.143(63.46)
2	0	0	28.22	1.968(58.26)
		1	30.56	1.986(58.81)
		2	31.29	2.001(59.26)
		3	31.55	2.015(59.67)
	1	0	29.12	2.713(80.33)
		1	31.59	2.732(80.88)
		2	32.33	2.747(81.33)
		3	32.60	2.760(81.73)
3	0	0	28.22	2.585(76.53)
		1	30.54	2.603(77.08)
		2	31.29	2.618(77.53)
		3	31.55	2.632(77.93)
	1	0	29.12	3.330(98.59)
		1	31.60	3.348(99.14)
		2	32.33	3.364(99.60)
		3	32.61	3.377(100.00)

where MSD is an acronym of mean squared difference and N is the number of pixels in the frame, and O_i and R_i are the intensity value of the original and the reconstructed frame. Note that the video coding system was set to the variable bit rate mode where its quantization parameter was fixed over the whole video sequence. The overall distortion data were measured in PSNR by averaging over 100 P-frames, using five video sequences, including *Carphone, Miss America, Foreman, Salesman*, and *Claire*.

IV. EXPERIMENTAL RESULTS

Based on the data in Table III, we searched optimal operating modes. Given the computational constraints $C_{\max}$, we were able to find optimal operating points by solving the optimization problem given in (2) and (3). We used two approaches, exhaustive search and the Lagrangian multiplier method. Note that our goal here was to find control variables s_1, s_2, and s_3, to maximize the cost function of the optimization problem, since we dealt with the overall distortion in PSNR.

Let $P_i, i = 0, \ldots, N-1$ represent an optimal operating point where N is the number of total optimal points by a search pro-cess. Using an exhaustive search, 11 optimal operating points were found and identified with P_0 to P_{10} in Fig. 5(a). Their control parameters are the same as follows: $(0,0,0), (0,0,1), (0,0,2), (0,0,3), (1,0,3), (0,1,1), (0,1,2), (0,1,3), (1,1,3), (2,1,3), (3,1,3)$, respectively. However, as shown in Table IV, the Lagrangian method, detected only 8 optimal operating points. Optimal operating points not located on the convex hull curve are not detected [28]. This

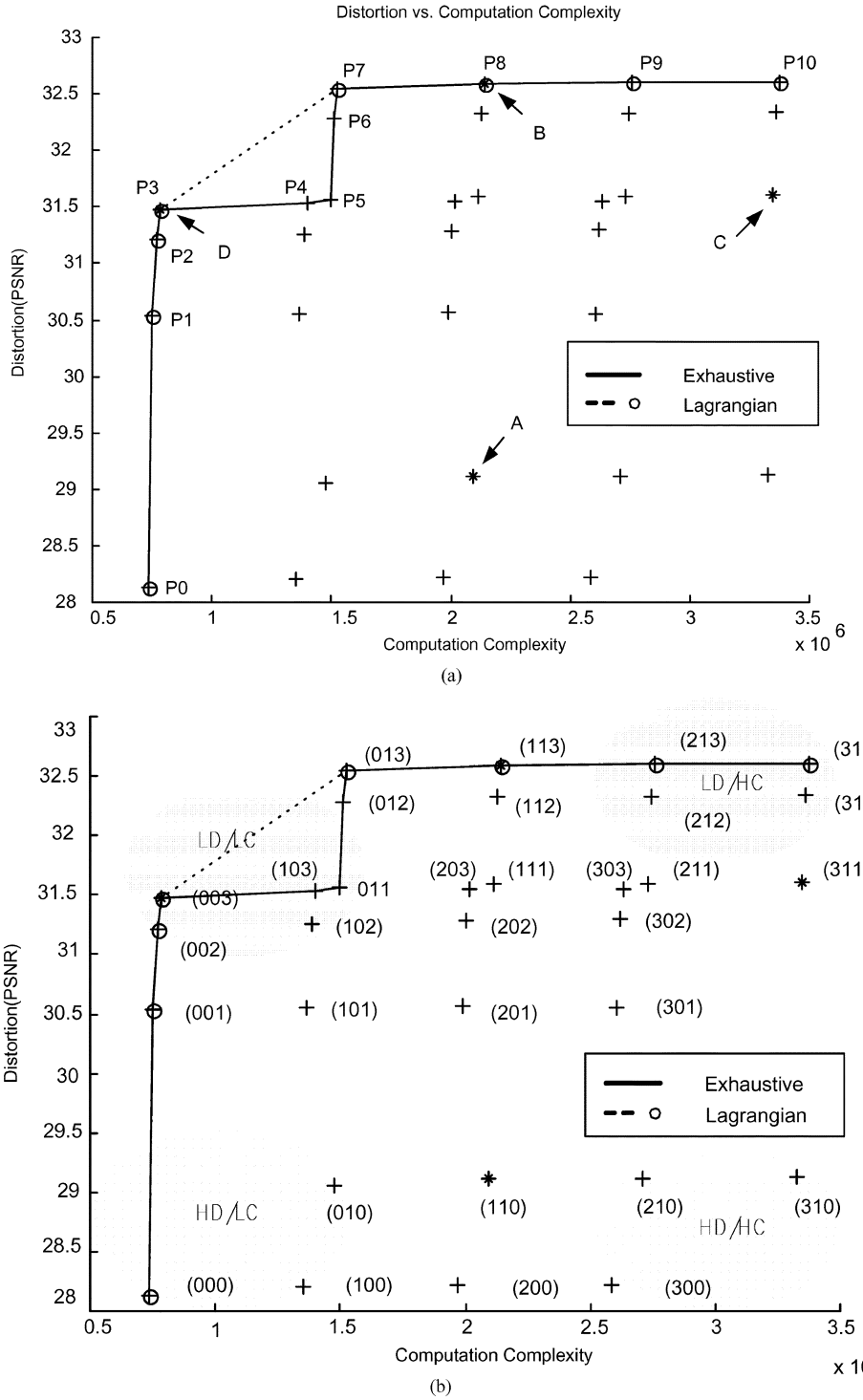


Fig. 5. Optimal operating modes found through exhaustive search over the real-measured C-D (PSNR) data with test video sequences. (a) Optimal operating modes. (b) Control parameters.

is shown graphically in Fig. 5(a), where optimal operating points are drawn with a solid line, and a dotted line corresponds to an exhaustive search and the Lagrangian multiplier method, respectively. Fig. 5 also demonstrates how important it is, from an overall system performance point of view, to select optimal operating modes among control variables. Note that four operating modes A, B, C, and D are identified using the marker "*" in the figure, whose control parameters are, respectively given as follows: $(1,1,0)$, $(1,1,3)$, $(3,1,1)$, and $(0,0,3)$.

Operating modes $C(3,1,1)$ and $D(0,0,3)$ have similar average PSNR distortions, but significant difference in complexities requiring 3.3×10^6 and 0.8×10^6 operations, respectively. Operating modes $A(1,1,0)$ and $B(1,1,3)$ have similar complexities concerning 2.1×10^6 operations, but a 3.48 dB difference in PSNR performance. This indicates that more computations do not necessarily perform better in an overall computation complexity space, which consists of combinations of all individual control variables. As expected, selecting optimal values

(a) (b)

Fig. 6. Comparison in subjective quality for two modes, "A" and "B" of Fig. 5 requiring similar computational complexity: the 6th frame, inter-coding, and $\mathrm{QP} = 13$ in the sequence "*Carphone*."

TABLE IV
OPTIMAL OPERATION MODES FOUND THROUGH LAGRANGIAN METHOD WHERE THE GIVEN COMPUTATIONAL COMPLEXITY IS CONTROLLED BY LAGRANGIAN MULTIPLIER λ OVER C-D DATA

Lagrangian Multiplier, λ	Operation Mode		
	ME s_1	I/H s_2	DCT s_3
0.000	3	1	3
0.001	2	1	3
0.009	1	1	3
0.020	0	1	3
0.454	0	0	3
7.434	0	0	2
18.350	0	0	1
77.187(∞)	0	0	0

of the control variables significantly influences the system's overall performance.

To demonstrate a comparison in the subjective performance, two sample video clips are shown in Fig. 6, where the subjective quality is clearly distinct between two operating modes, $A(1,1,0)$ and $B(1,1,3)$ of Fig. 5(a), closely located about 2.1×10^6 in the complexity axis. From this example, it is evident that the C-D optimal mode decision significantly affected the subjective performance of the video coding system.

In Fig. 5(b), there are four regions classified according to the complexity and the distortion as follows: HD/LC (High distortion and low complexity), HD/HC (high distortion and high complexity), LD/LC (low distortion and low complexity), and LD/HC (low distortion and high complexity). As shown in the figure, two regions HD/LC and LD/LC require low complexities and locate down and up in the left. On the other hand, HD/HC and LD/HC require high complexity and locate up and down in the right, respectively. Looking into the control parameters of modes and comparing one another located in different regions, it turns out that ME significantly influences the overall complexity, while DCT and H/I influence the overall distortion more than ME relatively.

A. Adaptive Mode Control

Video sequences have variations in characteristics including motion. This means that optimal operating modes defined by coding parameters change along with the changing video sequence. In other words, optimal C-D points should be controlled adaptively to achieve better performance. The adaptive control approach in regard to the operating modes is implemented and compared to the fixed approach. For the fixed method in the operating model control, the optimal control parameters given by (s_1, s_2, s_3) are searched in the initialization of the video encoding, under the given computational constraint, $C_{\max}$. These selected control parameters are used for all video frames and there is no update of the control parameters through whole video sequences. For the adaptive approach, however, the optimal control parameters $(s_1, s_2, s_3)_{t+1}$ for the next frame $t+1$ are searched iteratively after encoding every frame based on the C-D data, whose data entry is updated with the distortion of control parameters $(s_1, s_2, s_3)_t$ at the current frame t.

Basically, this adaptive scheme arises from the fact that the frame distortion varies through the entire video sequence. The update equation for the new optimal mode in the adaptive approach is given below

$$(s_1, s_2, s_3)_{t+1} = \min_{[s_1,s_2,s_3]\in S_1\times S_2\times S_3} D_t(s_1, s_2, s_3) \quad \text{subject to } C(s_1, s_2, s_3) \le C_{\max}. \tag{11}$$

where $(s_1, s_2, s_3)_{t+1}$ are the optimal control parameters for the frame $t+1$ and $D_t(s_1, s_2, s_3)$ is the distortion data in the C-D table, whose data entry is updated using the distortion of control parameters $(s_1, s_2, s_3)_t$ at the current frame t. In more detail, the algorithm of the adaptive mode control is described in the following steps.

Step 1) Let the computational constraint $C_{\max}$ be given, and $(s_1, s_2, s_3)_0 = (2, 1, 2)$ is set for the I-frame coding in the first frame. Assume that the initial

TABLE V
PERFORMANCE COMPARISON BETWEEN THE FIXED AND THE ADAPTIVE CONTROL OF THE OPERATING POINT, (S_1, S_2, S_3) WITH VIDEO SEQUENCES USED IN THE MODEL ESTIMATION

Constraint Control variable ρ = 0.8, $C_{\max}$ =2701908	**Complexity (Instructions)**		**Distortion (PSNR)**		**Rate (Bits)**	
	Fixed (s_1, s_2, s_3) = (1, 1, 3)	**Adaptive**	**Fixed**	**Adaptive**	**Fixed**	**Adaptive**
Carphone	2143449	2043211(0.95)	31.74	31.71	1908	1924(1.01)
Miss America	2143449	2143449(1.00)	35.86	35.86	674	674(1.00)
Foreman	2143449	1917978(0.89)	30.49	30.43	2704	2717(1.01)
Salesman	2143449	1901267(0.89)	30.23	30.18	948	958(1.01)
Claire	2143449	2143449(1.00)	34.61	34.61	665	665(1.00)

C-D data table, as given in Table III, is available by pre-processing off-line.

Step 2) Encode the first frame in the I-frame mode using control parameters initially given $(s_1, s_2, s_3)_0 = (2, 1, 2)$.

Step 3) Optimal control parameters $(s_1, s_2, s_3)_t$ for frame t are searched from the C-D table. Encode in P-frame mode from the second frames.

Step 4) Calculate the distortion of $D_t(s_1, s_2, s_3)$ at the frame t corresponding to the control parameters $(s_1, s_2, s_3)_t$. Update the C-D table entry with the distortion $D_t(s_1, s_2, s_3)$.

Step 5) Increase the frame number $t = t + 1$ and jump back to Step 3. Repeat Steps 3–5 until the end of sequence.

In following comparisons of rate performance, the video coding system was set to the variable bit rate mode, where its quatization parameter was fixed over whole video sequence, since the distortion model parameters were estimated with the fixed quantization parameter. Table V shows experimental results with the fixed and the adaptive control of operating modes. The same five video sequences involved in the estimation process of the distortion parameters in the C-D model were used for the experiment. All 100 frames were coded and averaged, where the first frame was intra-coded and other following frames were inter-coded with the quantization parameter (QP) set to 13.

Let a variable $\rho \in \{0.0, \ldots, 1.0\}$ denote a weighting factor to the computation complexity of the system represented by the maximum values of operation modes. The computational constraint value $C_{\max}$ is relative to the maximum system complexity and derived by multiplying it with the constraint control variable ρ. It is shown in the table that $C_{\max}$ is controlled by the constraint control variable ρ. This can be calculated by multiplying the control variable ρ to the maximum complexity of the operation mode, (s_1, s_2, s_3), in the C-D model. This calculation can be given as

$$C_{\max} = \rho \times C(s_{1m}, s_{2m}, s_{3m}) \tag{12}$$

where ρ is the constraint control variable and $C(s_{1m}, s_{2m}, s_{3m})$ is the complexity for the operating mode (s_{1m}, s_{2m}, s_{3m}), having the maximum complexity in the C-D model. The maximal complexity mode (s_{1m}, s_{2m}, s_{3m}) corresponds to $(3, 1, 3)$ in the C-D model shown in Table III. In Table V, as an example, the constraint control variable was set to $\rho = 0.8$.

It is clearly proven in the table that the adaptive control works better with an active sequence, having more motions than with other silent sequences. For example, *Carphone, Foreman*, and *Salesman* sequences showed better performance with an adaptive control feature, while other silent sequences such as *Miss America* and *Claire* showed no significant difference between the fixed and the adaptive control methods. With the sequences *Foreman* and *Salesman*, the computational complexity saved about 11% using the adaptive control, while it incurs degradation, less than 0.06 dB. We also investigated how C-D optimization methods affect total bit rates. Generally, the bit rate is related to the coding efficiency, including motion estimation. As shown in the table, there is no significant difference of bit rate between the two control modes. Fig. 7 shows complexity changes according to the operating modes detected adaptively by the C-D optimization algorithm. In the figure, operating modes $P_i, i \in \{0, \ldots, N-1\}$ are represented with the control parameters (s_1, s_2, s_3). Complexity numbers corresponding to the operating modes are the same as ones shown in Table III. For example, the first 10 operating modes $P_i, i \in \{0, \ldots, 9\}$ are given as follows, respectively:

TABLE VI
PERFORMANCE COMPARISON BETWEEN THE FIXED AND THE ADAPTIVE CONTROL IN THE OPERATING POINT, (S_1, S_2, S_3) WITH OTHER VIDEO SEQUENCES NOT USED IN THE MODEL ESTIMATION

Constraint Control Factor ρ **= 0.8,** $C_{\max}$ **=2701908**	**Complexity (Instructions)**		**Distortion (PSNR)**		**Rate (Bits)**	
	Fixed (S_1, S_2, S_3) **= (1, 1, 3)**	**Adaptive**	**Fixed**	**Adaptive**	**Fixed**	**Adaptive**
Container	2143449	1820380(0.85)	30.93	30.90	991	990(1.00)
Grandma	2143449	2009672(0.94)	32.02	32.01	582	577(0.99)
Mothr_dautr	2143449	2068504(0.97)	31.50	31.49	1335	1338(1.00)
News	2143449	1742469(0.81)	30.79	30.75	1508	1545(1.03)
Suzie	2143449	2124939(0.99)	33.02	33.02	1664	1667(1.00)

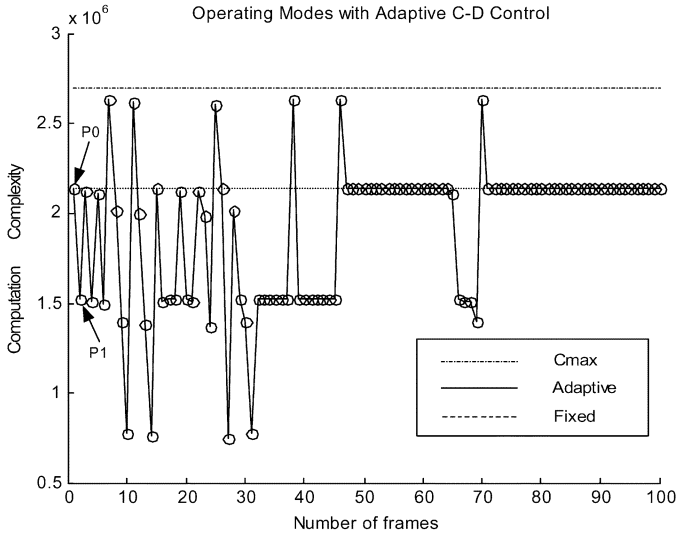


Fig. 7. Operating mode found by adaptive C-D control in the sequence "*Foreman*."

$(1,1,3),(0,1,3),(1,1,2),(0,1,2),(1,1,1),(0,1,1),(3,0,3),$ $(2,0,3),(1,0,3),(0,0,3),(3,0,2),(2,0,2),(1,0,2),(0,0,2)$.

Note that the distortion parameters of the C-D model were estimated using five video sequences. It would be interesting to investigate how much more effective the estimated model parameters would be with other video sequences not involved in the model estimation process. Table VI shows experimental results using the following five video sequences: *Container, Grandma, Mothr_dautr, News*, and *Suzie*. The quantization parameter QP was fixed to 13. The first frame was intra-coded and those that followed were inter-frame coded. For the sake of comparison, the results were obtained by averaging over 100 frames. As shown in the table below, the C-D model works well, even with other video sequences not considered in the model estimation process. With active sequences such as *Container* and *News*, the adaptive control method performed best in the C-D optimization.

With the various sequences above, computation reductions were obtained up to 19% compared to the fixed method, while the degradations of the reconstructed video were less than 0.05 dB. Furthermore, there was no significant difference between the adaptive and the fixed methods in rate performance. Based on these experimental results, it is evident that the estimated C-D model parameters are accurate enough to be applied to most video sequences, regardless of their motion.

V. CONCLUSION

The performance of a computationally configurable video coding scheme with respect to computational C-D, has been analyzed. The proposed coding scheme consists of three coding modules: motion estimation, sub-pixel accuracy, and DCT pruning, whose control variables can take several values, leading to significantly different performance for the coding. This analysis confirms that a configurable video coding system where the control parameters are chosen optimally leads to better performance. To evaluate the performance of proposed scheme according to input video sequences, we applied video sequences other than those involved in the process of model parameter estimation, and showed that the model parameters are accurate enough to be applied regardless of the type of input video sequences.

Furthermore, an adaptive scheme to find the optimal control parameters of the video modules was introduced and compared

with the fixed. The adaptive approach was proven to be more effective with active video sequences rather than with silent video sequences.

ACKNOWLEDGMENT

The authors would like to thank H. Jeon, T. Reino Huitica, and Z. Zhang for their technical discussion and comments in realizing the proposed idea and writing the paper.

REFERENCES

[1] A. Ortega and K. Ramchandran, "Rate distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[2] G. J. Sullivan and T. Wiegand, "Rate distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[3] B. Girod, "Rate constrained motion estimation," in *Proc. SPIE Conf. Visual Commun. Image Process.*, vol. 2308, 1994, pp. 1026–1034.

[4] G. M. Schuster and A. K. Katsaggeelos, "Fast efficient mode and quantizer selection in the rate distortion send for H.263," in *Proc. SPIE Conf. Visual Commun. Image Process.*, Mar. 1996, pp. 784–795.

[5] K. Lengwehasatit and A. Ortega, "Rate complexity distortion optimization for quad tree based DCT," in *Proc. Int. Conf. Image Processing*, vol. 3, 2000, pp. 821–824.

[6] V. Goyal and M. Vetterli, "Computation distortion characteristics of block transform coding," in *Proc. ICASSP*, vol. 4, Munich, Germany, Apr. 1997, pp. 2729–2732.

[7] I. Ismaeil, A. Docef, F. Kossentini, and R. Kreidieh, "A computation-distortion optimized framework for efficient DCT-based video coding," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 298–310, Sep. 2001.

[8] *Draft Recommendation H.263*, Apr. 7, 1995.

[9] J. R. Jain and A. K. Jain, "Displacement measurement and its application in inter frame image coding," *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799–1808, Dec. 1981.

[10] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated inter frame coding for video conferencing," in *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, Nov.-Dec. 1981, pp. G5.3.1–5.3.5.

[11] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, Aug. 1998.

[12] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[13] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[14] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, 2nd ed. Norwell, MA: Kluwer, 1997.

[15] H. Fujiwar, "An all-ASIC implementation of a low bit-rate video codec," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 123–134, Jun. 1992.

[16] K. Guttag, R. J. Cove, and J. R. Van Aken, "A single chip multprocessor for multimedia: The MVP," *IEEE Comput. Graph. Applicat.*, vol. 12, no. 6, pp. 53–64, Nov. 1992.

[17] C. G. Zhou, "MPEG video decoding with the UltraSPARC visual instruction set," in *IEEE Dig. Papers COMPCON*, Mar. 1995, pp. 470–477.

[18] B. Furht, J. Greenberg, and R. Westwater, *Motion Estimation Algorithms for Video Compression*. Norwell, MA: Kluwer, 1997.

[19] P. Kuhn, *Algorithms, Complexity Analysis and VLSI Architectures for MPEG4 Motion Estimation*. Norwell, MA: Kluwer, 1999.

[20] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1243–1245, Dec. 1984.

[21] K. R. Rao and P. Yip, *Discrete Cosine Transform—Algorithms, Advantages, Applications*. New York: Academic, 1990.

[22] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Trans. IEICE*, vol. E-71, no. 11, pp. 1095–1097, Nov. 1988.

[23] A. N. Skodras, "Fast discrete cosine transform pruning," *IEEE Trans. Signal Process.*, vol. 42, no. 7, pp. 1833–1837, Jul. 1994.

[24] Z. Wang, "Pruning the fast discrete cosine transform," *IEEE Trans. Commun.*, vol. 39, no. 5, pp. 640–643, May 1991.

[25] S. C. Chan and K. L. Ho, "A new two-dimensional fast cosine transform algorithm," *IEEE Trans. Signal Process.*, vol. 39, no. 2, pp. 481–485.

[26] G. M. Schuster and A. K. Katsaggelos, "A theory for the optimal bit allocation between displacement vector field and displaced frame difference," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 9, pp. 1739–1751, Dec. 1997.

[27] Y. Yang and S. S. Hemami, "Generalized rate distortion optimization for motion compensated video coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 942–955, Sep. 2000.

[28] G. M. Schuster and A. K. Katsaggelos, *Rate Distortion Based Video Compression*. Norwell, MA: Kluwer, 1997.

[29] C. Y. Hsu and A. Ortega, "A Lagrangian optimization approach to rate control for delay-constrained video transmission over burst error channels," in *Proc. ICASSP*, Seattle, WA, May 1998, pp. 2989–2992.

[30] A. Ortega, "Optimal bit allocation under multiple rate constraints," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 1996, pp. 2989–2992.

[31] J. J. Chen and D. W. Lin, "Optimal bit allocation for video coding under multiple constraints," in *Proc. IEEE Int. Conf. Image Process.*, 1996, pp. 349–358.

[32] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate distortion sense," *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 160–175, Apr. 1993.

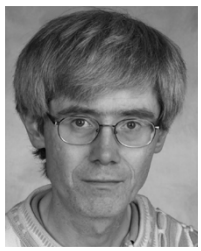
[33] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.

[34] G. M. Schuster and A. K. Katsaggelos, "An optimal quad tree based motion estimation and motion based interpolation scheme for video compression," *IEEE Trans. Image Process.*, vol. 7, no. 11, pp. 1505–1523, Nov. 1998.

[35] D. N. Kwon, P. Driessen, and P. Argathoklis, "Performance and computational complexity optimization in a configurable video coding system," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2003, pp. 2086–2089.

David Nyeongkyu Kwon (SM'00) received the B. S. degree from Han-Kuk Aviation University, Seoul, Korea, in 1998, and the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), Seoul, Korea, in 1990, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical and computer engineering from the University of Victoria, Victoria, BC, Canada.

From 1990 to 1994, he was a Research Engineer in the RADAR Division of Agency of Defense Development (ADD), Seoul, Korea. His research interests include multimedia signal and video processing, multimedia transmission over wire/wireless network, multimedia ASIC design and implementation.



Peter F. Driessen (M'89–SM'93) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 1981.

He has worked with various companies in Vancouver on several projects related to wireless data transmission and modem chip design. Since 1986, he has been at the University of Victoria, Victoria, BC, Canada, where he is now Professor in the Department of Electrical and Computer Engineering. He was on sabbatical leave at AT&T Bell Laboratories, Holmdel, NJ, during the academic year 1992–1993, and at AT&T Laboratories-Research, Red Bank, NJ, during the academic year 1999–2000. His research interests are in aspects of wireless communications systems, audio signal processing and streaming multimedia over packet networks. He has served as an Editor for *IEEE Personal Communications Magazine* from 1997 to 1999 and as an Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Wireless Communications Series (now IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS) from 1999 to the present.

Andrea Basso received the M.Sc. degree in electrical and computer engineer from the University of Trieste, Treiste, Italy, and the Ph.D. degree in video processing from EPFL, Lausanne, Switzerland.

From April 1989 to April 1990, he was a Visiting Student at the IRST-Trento, Italy. From 1990 to 1995, he was with the Signal Processing Laboratory, EPFL, Lausanne, Switzerland. From 1995 to 1996, he was with the Telecommunication Laboratory (TCOM), EPFL, as Head of the Multimedia Communication Team. In 1995, he was a Visiting Research Associate in the Multimedia Communications Group, Electrical Engineering Department, Stanford University, Stanford, CA. During the fall of 1995, he was a Consultant at AT&T Bell Labs, Holmdel, NJ, in the Visual Communications Department. From 1997 to 1999, he was with AT&T Labs—Research, Florham Park, NJ, in the Speech and Video Technology Research Group as a Senior Technical Staff Member. From January 2000 to 2002, he was with AT&T Labs—Research in the Broad-band Telecommunications Laboratory as a Principal Technical Staff Member. He is currently a Principal Architect and Technical Director with NMS Communications, Red Bank, NJ. He serves on numerous editorial boards in the area of multimedia and networking. He is author or coauthor of 50 papers and three books. He holds eight patents. His current research interests include still and sequence image representation and coding, real time communications, scalability and interworking aspects of multimedia with particular focus on quality of service, and inter- and intra- media synchronization.

Panajotis Agathoklis (M'81-SM'88) received the Dipl.Ing. in electrical engineering and the Dr.Sc. Tech. degree from the Swiss Federal Institute of Technology, Zurich, Switzerland, in 1975 and 1980, respectively.

From 1981 until 1983, he was with the University of Calgary as a Post-Doctoral Fellow and part-time Instructor. Since 1983, he has been with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada, where he is currently Professor. He has been member of the Technical Program Committee of many international conferences and has served as the program chair of the 1991 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. His fields of interests are in digital signal processing, system theory and stability analysis.

Dr. Agathoklis received a NSERC University Research Fellowship (1984–1986) and Visiting Fellowships from the Swiss Federal Institute of Technology (1982, 1984, 1986, and 1993), from the Australian National University (1987) and the University of Perth, Australia (1997). He was Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in 1990–1993.