Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 47921, 10 pages
doi:10.1155/2007/47921

Research Article

A Complexity-Aware Video Adaptation Mechanism for Live Streaming Systems

Meng-Ting Lu,[1] Jason J. Yao,[1] and Homer H. Chen[2]

[1] *Department of Electrical Engineering, Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan*
[2] *Department of Electrical Engineering, Graduate Institute of Communication Engineering, and Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan*

Received 3 October 2006; Accepted 21 March 2007

Recommended by Alex Kot

The paradigm shift of network design from performance-centric to constraint-centric has called for new signal processing techniques to deal with various aspects of resource-constrained communication and networking. In this paper, we consider the computational constraints of a multimedia communication system and propose a video adaptation mechanism for live video streaming of multiple channels. The video adaptation mechanism includes three salient features. First, it adjusts the computational resource of the streaming server block by block to provide a fine control of the encoding complexity. Second, as far as we know, it is the first mechanism to allocate the computational resource to multiple channels. Third, it utilizes a complexity-distortion model to determine the optimal coding parameter values to achieve global optimization. These techniques constitute the basic building blocks for a successful application of wireless and Internet video to digital home, surveillance, IPTV, and online games.

1. INTRODUCTION

Multimedia streaming is one of the most challenging services over the Internet, for which bandwidth is a primary constraint. For live streaming of multiple channels, the computational resource also becomes a critical issue. Our objective of this work is to develop a video adaptation mechanism to control the allocation of both bandwidth and computational resources for live streaming.

The problem of resource-constrained video coding has been the focus of research for the past several decades. For bandwidth constraint, various rate-distortion (R-D) models [1–4] have been proposed to deal with the tradeoff between information rate and distortion. For computational resource constraint, many algorithms have been developed to regulate the complexity of an encoder. Tai et al. [5] reported a software-based computation-aware scheme that terminates the searching process once a specified amount of computation has been reached. Chen et al. [6] developed an adaptive search strategy to find the best block matching in a computation-limited environment. Zhao and Richardson [7] designed adaptive algorithms for DCT and motion estimation to reduce the complexity of each function and maintain the computational cost at the target level. Zhong and Chen [8] proposed to dynamically regulate encoding complexity to achieve real-time performance and maximize coding efficiency.

Recently, joint complexity-rate-distortion constraints have been considered for video coding. He et al. [9] developed a power-rate-distortion (P-R-D) model to maximize the video quality subject to an energy constraint for wireless video communications. Stottrup-Andersen et al. [10] designed an operational method for optimizing integer motion estimation in real-time H.264 encoding by considering the tradeoff between rate distortion and complexity.

On the other hand, van der Schaar et al. [11–14] proposed a generic rate-distortion-complexity model to estimate the complexity of image and video decoding algorithms running on various hardware architectures. Based on the model, the receivers can negotiate with the server a desired complexity level with their available computational resources.

All the above schemes handle either the optimization of video quality of a single encoder or the negotiation with receivers to determine the optimal transmission policy based on the estimated decoding complexity. However, for live

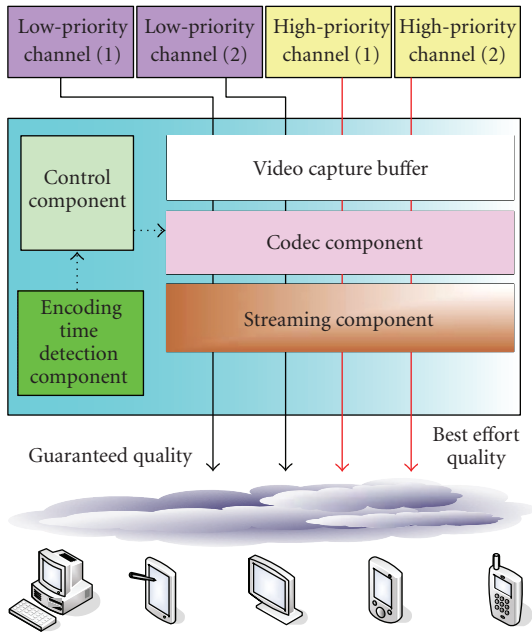


Figure 1: Complexity-aware live streaming server architecture.

streaming of multiple channels, which is the main scenario considered here, the key task is not only to optimize the video quality of a single channel but also to decide how to allocate appropriate computational resource to each stream [15–18]. The computation overhead of the allocation process must be small enough to meet the real-time constraint of live streaming. This is particularly important for resource-constrained communications.

In this paper, we propose the design of a complexity-aware live streaming system for multiple channels. Our system solves the problem of resource allocation by establishing a complexity-distortion (C-D) model through clustering. The C-D model helps the resource allocation module to predict the encoding characteristics of video sequences and minimize the sum of distortions to achieve global optimization. To reduce the execution overhead of resource allocation, we formulate the optimization problem as a linear programming problem with piecewise linear approximation. After resource allocation, the complexity control module optimizes the video quality of each stream individually at the block level. The allocation process is done in real time without affecting the performance of video encoding.

The rest of the paper is organized as follows. Section 2 presents the architecture of the live streaming server and the design of the control mechanism. Section 3 describes the complexity control mechanism that adjusts the encoding complexity of each frame at the block level. Section 4 describes the analysis and clustering based on the encoding characteristics of sequences. Section 5 discusses how to determine the available resource for the subsequent frames to be processed, and Section 6 describes the mechanism of resource allocation to achieve global optimization. Simulation results are given in Section 7 and the conclusion is in Section 8.

2. PROPOSED LIVE STREAMING SYSTEM

Figure 1 shows the architecture of the proposed live streaming system which consists of five major function blocks: encoding time detection, control, codec, streaming component, and video capture buffer. The encoding time detection component records the consumed CPU time of the codec component, while the control component allocates the computational resource based on the information provided by the encoding time detection component. The codec component encodes input video frames temporarily stored in the video capture buffer with parameter values determined by the control component, and the streaming component transmits the encoded videos with RTP/RTCP [19]. The time that a video frame spends in the video capture buffer and codec component is called the encoding delay, which is to be controlled in our system. In this architecture, the input videos are encoded differentially according to their priorities, the available bandwidth, as well as the computational resource of the server. Channels with higher priority are delivered with guaranteed quality while channels with lower priority are served with best-effort quality.

The control component is the most sophisticated part in our streaming server architecture. Figure 2 shows the block diagram of its three modules: available resource calculation (ARC), resource allocation (RA), and block-based complexity control (BCC). The ARC module determines the available computational resource, $T_{\text{available}}$, for subsequent frames by considering the actual encoding time, T_{actual}, of the current frame and the accumulation error, T_D. The RA module allocates $T_{\text{available}}$ to all channels according to their priorities in a globally optimized way. The BCC module calculates the encoding parameter values to minimize the distortion of each channel under the resource allocation constraint by the C-D model developed in this paper. Table 1 defines the symbols used in this paper.

3. BLOCK-BASED COMPLEXITY CONTROL

The complexity control mechanism minimizes the distortion of each channel by varying the values of encoding parameters [5–8]. For example, Tai et al. [5] try to find the best motion estimation mechanism under the computational constraint. We choose to adjust the encoding complexity at the block level for better control. To determine the parameters for encoding complexity adjustment, we model the factors affecting encoding complexity by

$$C_{\text{Total}} = C_{\text{ME}} + C_{\text{TQ}} + C_{\text{ENC}} + C_X, \tag{1}$$

where C_{Total} denotes the total complexity, C_{ME} the computational complexity of motion estimation, C_{TQ} the complexities of transform coding and quantization, C_{ENC} the complexity of entropy coding, and C_X the overhead complexity not controlled by the encoder. C_X includes the complexity of CPU context switch and memory access, which vary for

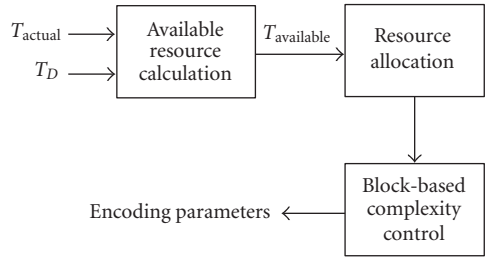


Figure 2: Control component block diagram.

different computers and operating systems. Because C_X is uncontrollable, we do not consider it in our system and just treat it as the reduction of the computational resource. To further model the other terms in (1), C_{ME} is expressed as

$$C_{ME} = \lambda_{ME1} C_{SAD(16\times16)} + \lambda_{ME2} C_{SAD(8\times8)} + C_{HP}, \quad (2)$$

where C_{SAD} denotes the complexity of one SAD operation, λ_{ME1} the number of 16×16 SAD operations, λ_{ME2} the number of 8×8 SAD operations, and C_{HP} the complexity of half-pel refinement. Note that C_{TQ} is determined by several factors: the computational complexities of DCT, IDCT, quantization, and dequantization. It is known that the output of an all-zero macroblock after transform coding and quantization is still an all-zero macroblock, and that the reconstructed macroblock is exactly the reference macroblock. This property is utilized to reduce the computational complexity by skipping the coding of all-zero macroblocks. Therefore, C_{TQ} is only determined by the complexity of the coding of nonzero macroblocks and is formulated as

$$C_{TQ} = \lambda_{TQ} C_{NZMB}, \quad (3)$$

where λ_{TQ} denotes the number of nonzero macroblocks and C_{NZMB} the computational complexity of the coding of a nonzero macroblock. The relation between the complexity of entropy coding and bit rate R is expressed as

$$C_{ENC} = RC_{Bit}, \quad (4)$$

where C_{Bit} denotes the computational complexity of entropy coding of a bit. The definition of encoding complexity factors in this paper is improved from the one in [9]. We consider additional factors about motion estimation with different block sizes and motion vector resolutions, which makes the resource allocation more flexible and accurate. Besides, the overhead complexity not controlled by the encoder is also considered.

Equations (1)–(4) indicate that the encoding complexity at a fixed bit rate is affected by the number of SAD operations M_{Total} and the number of nonzero macroblocks N_{NZMB} of each video frame. Therefore, M_{Total} and N_{NZMB} are used to control the encoding complexity. The number of SAD operations allocated to the ith macroblock M_i is then calculated by

$$M_i = M_{Total} \frac{SAD_i}{SAD_{Total}}, \quad (5)$$

Table 1: Nomenclature.

T_D	Accumulated sum of $(T_{actual} - T_{target})$
T_{actual}	Actual encoding time for the current frame
T_{target}	Target encoding time for each frame
$T_{available}$	Available encoding time for the subsequent frames
$T_{average}$	Average encoding time for the encoded frames
M_{Total}	Number of search points allocated to the current frame
N_{NZMB}	Number of nonzero macroblocks allocated to the current frame
$\mathbf{p_i}$	The vector $[M_{Total} \quad N_{NZMB}]^T$ for the ith channel
$D_i(\mathbf{p_i})$	Estimated distortion for the ith channel
$C_i(\mathbf{p_i})$	Average encoding time for the ith channel
w_i	The current operation point on the complexity-distortion curve for the ith channel
m_{W_i}	The slope of the approximated line of the complexity-distortion curve given the current operation point w_i
E_i	The representative encoding characteristics for the ith group
N_H	Number of high-priority channels
N_L	Number of low-priority channels

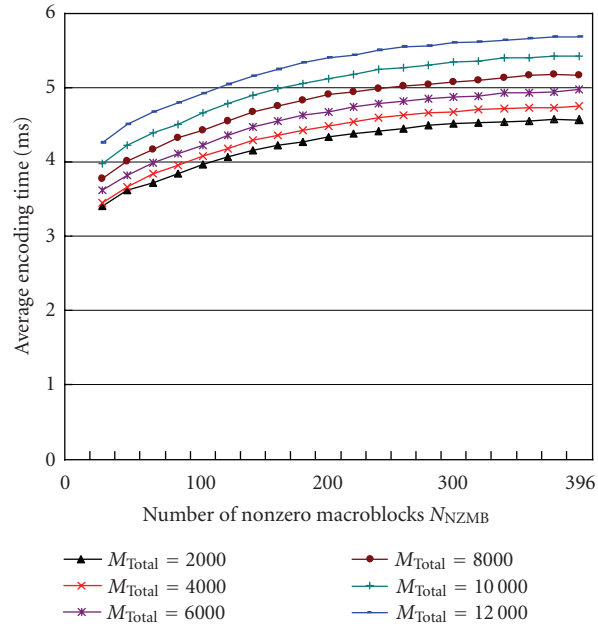


Figure 3: The average encoding time of the Football sequence for different numbers of nonzero macroblocks (N_{NZMB}) and SAD operations (M_{Total}).

where SAD_i is the SAD value of the collocated macroblock in the previous frame and SAD_{Total} is the sum of SAD values of the previous frame. Equation (5) utilizes the temporal relationship of residuals to allocate the number of SAD operations to each macroblock. If the residual of the ith macroblock in the previous frame is large, there is a high probability that the residual of the same macroblock in the current frame is also large. More SAD operations are allocated to

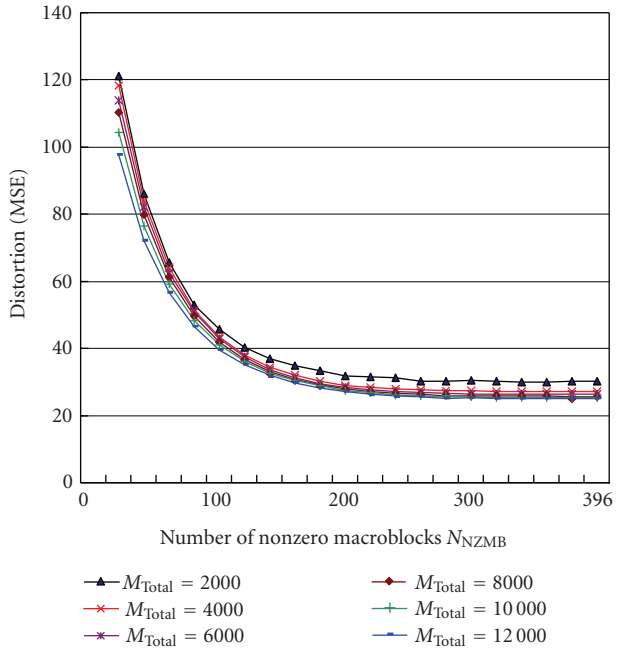


Figure 4: The distortions of the Football sequence for different numbers of nonzero macroblocks (N_{NZMB}) and SAD operations (M_{Total}).

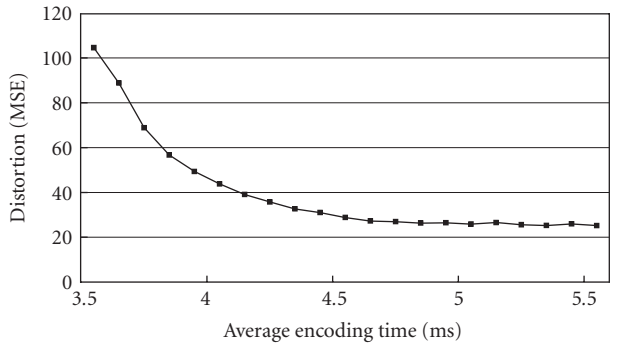


Figure 5: Minimum distortion versus average encoding time.

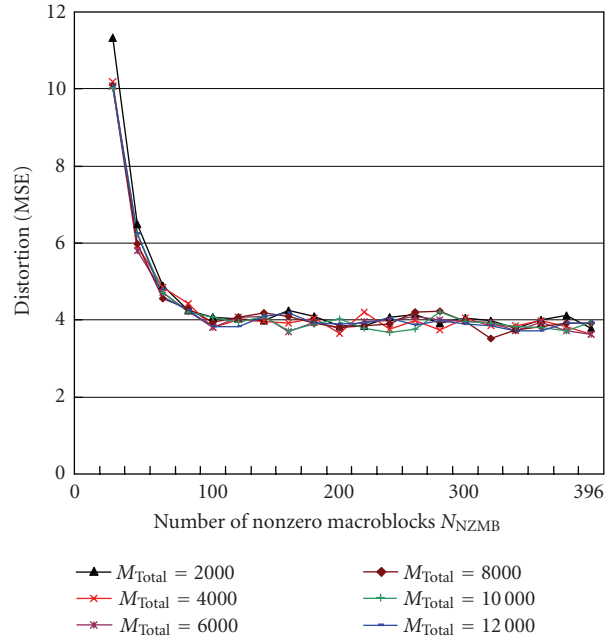


Figure 6: The distortions of the Weather sequence for different numbers of nonzero macroblocks (N_{NZMB}) and SAD operations (M_{Total}).

macroblocks with larger residual. After motion estimation, the residuals of the macroblocks are sorted in a descending order. The first N_{NZMB} macroblocks are coded as nonzero macroblocks while the remaining macroblocks are treated as all-zero macroblocks.

4. COMPLEXITY-DISTORTION MODELING

To develop the C-D model, simulations are performed on the BCC module to obtain the relationship between M_{Total}, N_{NZMB}, average encoding time $T_{average}$, and the distortion of a video sequence. In this paper, such a relationship is called the encoding characteristics of a video sequence, denoted as ECs. The ECs help us to predict $T_{average}$ and distortion before deciding the values of M_{Total} and N_{NZMB}. For the simulations in Figures 3–8, only I and P frames are used with I frame interval being 30 frames, and the bit rate is fixed at 1 Mbps. Figures 3–5 show the ECs of the CIF-sized Football sequence. Figure 3 illustrates the values of $T_{average}$ for different values of M_{Total} and N_{NZMB}, and Figure 4 shows the resulting distortion. In Figures 3 and 4, there exist many combinations of M_{Total} and N_{NZMB} for a target average encoding time T_{target}. By searching through all these combinations, the minimum distortion and corresponding values of M_{Total} and N_{NZMB} are obtained, and the results of minimum distortion versus T_{target} are shown in Figure 5.

ECs are different for each video sequence and unavailable for live streaming. As shown in Figures 4 and 6–8, the curves differ from one video sequence to another. Figures 6 and 7 illustrate that the maximum distortion of the Weather sequence is less than 12, while that of the Mobile sequence is more than 250. Moreover, as M_{Total} increases, the distortion of the Weather sequence remains nearly unchanged, while the distortion of the Mobile sequence decreases noticeably. However, some similarities exist among the video sequences in spite of the differences described above. From Figures 6 and 8, it is observed that the distortion and the curves are similar in the Weather and the Container sequences, which implies that it is feasible to predict the ECs of the Weather sequence based on the information of the Container sequence. Thus, we establish the complexity-distortion model by clustering the training video sequences into four groups. Let E_i denote the representative of the ECs of the ith group. For a new video sequence, the nearest E_i is determined by obtaining the encoding characteristics of the first several frames, and the video sequence is assigned to the ith group. Then,

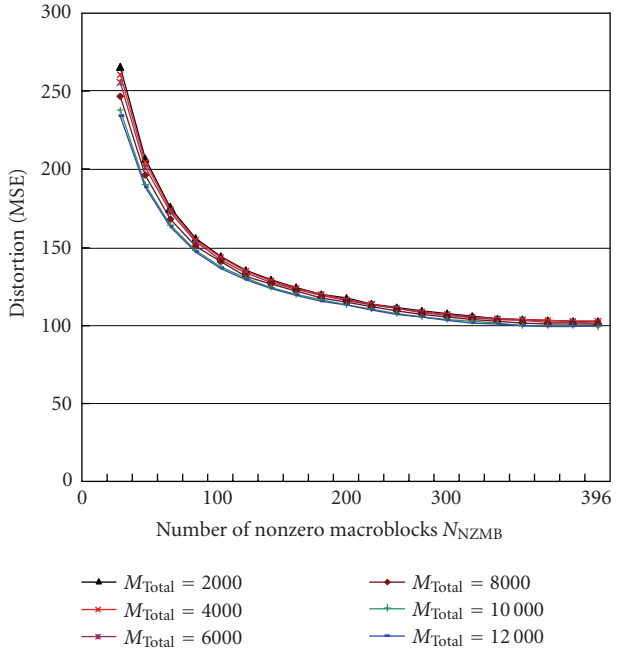


FIGURE 7: The distortions of the Mobile sequence for different numbers of nonzero macroblocks (N_{NZMB}) and SAD operations (M_{Total}).

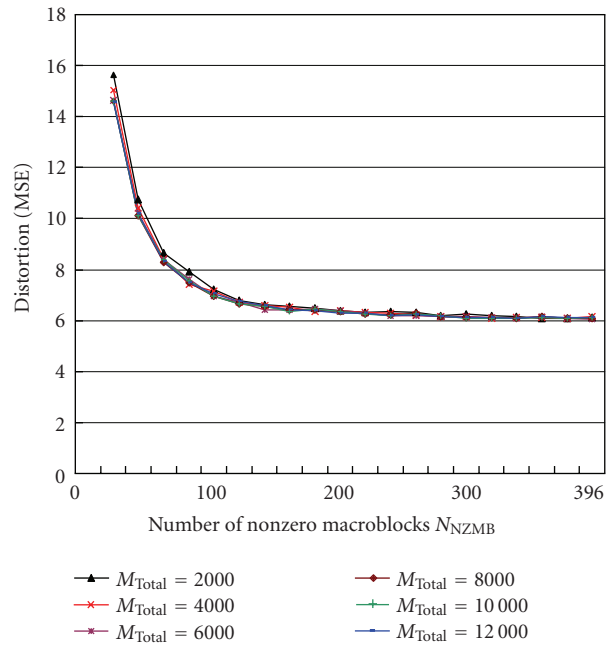


FIGURE 8: The distortions of the Container sequence for different numbers of nonzero macroblocks (N_{NZMB}) and SAD operations (M_{Total}).

TABLE 2: Results of clustering.

Cluster 0	Coastguard, Football, Tempete
Cluster 1	Bus, Canoa, Stefan
Cluster 2	Container, Dancer, Foreman, Hall monitor, Mother and daughter, Paris, Silent, Table, Weather
Cluster 3	Mobile

we can predict the ECs of the new video sequence according to E_i. Because our system determines the parameter values of the C-D model by finding the nearest neighbor, the complexity is much smaller than the C-D model in [9] which uses linear regression to estimate model parameter values from the statistics of previous frames.

In the clustering process, the ECs of the ith sequence are represented by a vector

$$\mathbf{V_i} = \left[\mathbf{T_i} \;\; \mathbf{D_i}^T\right], \tag{6}$$

where $\mathbf{T_i}$ denotes the values of $T_{average}$ and $\mathbf{D_i}$ the average distortion for different values of M_{Total} and N_{NZMB}. Furthermore, $\mathbf{T_i}$ is expressed as

$$\mathbf{T_i} = \left[T_{1,1} \;\; T_{1,2} \;\; \cdots \;\; T_{j,k} \;\; \cdots \;\; T_{20,5} \;\; T_{20,6}\right], \tag{7}$$

where $T_{j,k}$ denotes the value of $T_{average}$ when N_{NZMB} equals j multiplied by 20 and M_{Total} equals k multiplied by 2000. $\mathbf{D_i}$ is expressed as

$$\mathbf{D_i} = \left[D_{1,1} \;\; D_{1,2} \;\; \cdots \;\; D_{j,k} \;\; \cdots \;\; D_{20,5} \;\; D_{20,6}\right], \tag{8}$$

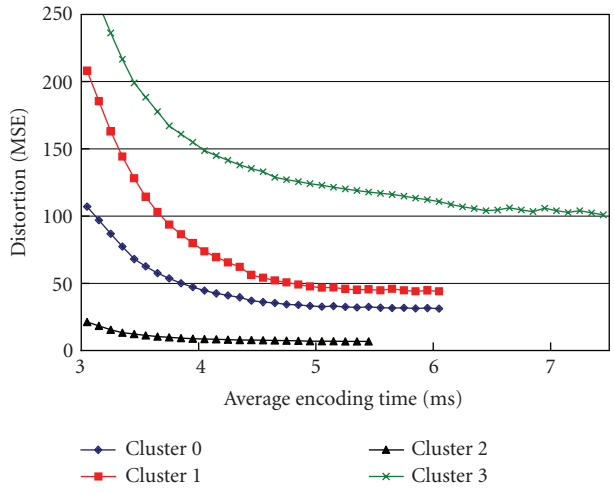


FIGURE 9: The relationship between average encoding time and distortion for the center of each cluster.

where $D_{j,k}$ denotes the average distortion when N_{NZMB} equals j multiplied by 20 and M_{Total} equals k multiplied by 2000. By applying the K-means algorithm [20–22] to the vectors of ECs, we obtain the clustering results shown in Table 2, and the relationship between $T_{average}$ and distortion for the representatives of the four groups shown in Figure 9. The complexity-distortion curves and the corresponding values of N_{NZMB} and M_{Total} form the C-D model, which helps us to predict the ECs of video sequences in each cluster. To check the validity of the C-D model, the complexity-distortion curves of video sequences and the curves in Figure 9 are

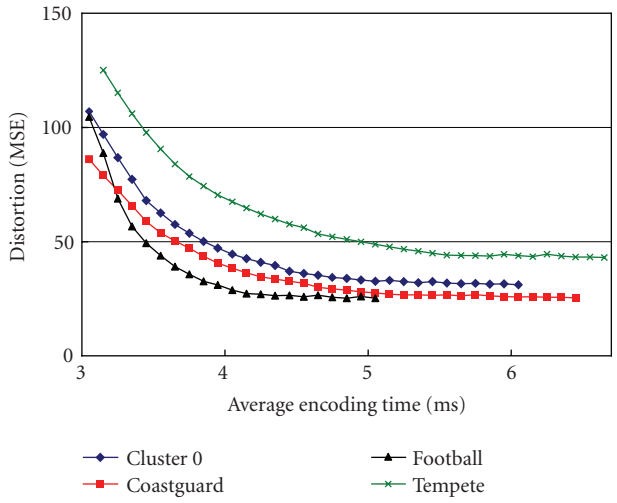


Figure 10: The relationship between average encoding time and distortion for video sequences in cluster 0.

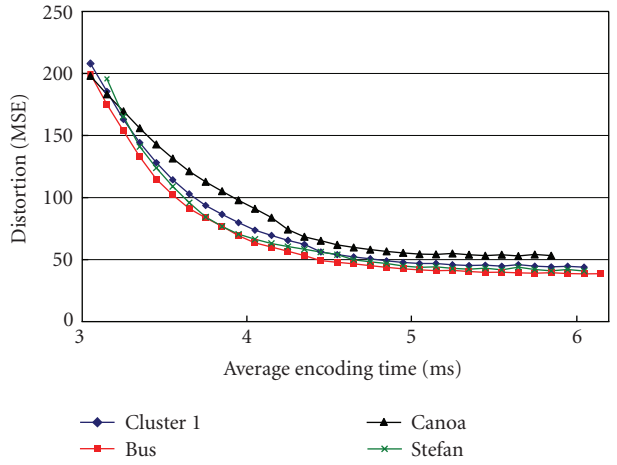


Figure 11: The relationship between average encoding time and distortion for video sequences in cluster 1.

Table 3: Average execution time to solve a linear programming problem.

Number of variables	3	4	5	6	7
Execution time (ms)	0.302	0.308	0.310	0.331	0.335

compared in Figures 10–13, which demonstrates that the curves of video sequences are very close to those of the representatives, implying that the complexity of each video sequence can be adjusted based on the C-D model.

5. AVAILABLE RESOURCE CALCULATION

The ARC module computes the available resource for the subsequent frames based on the values of T_D, T_{actual}, and

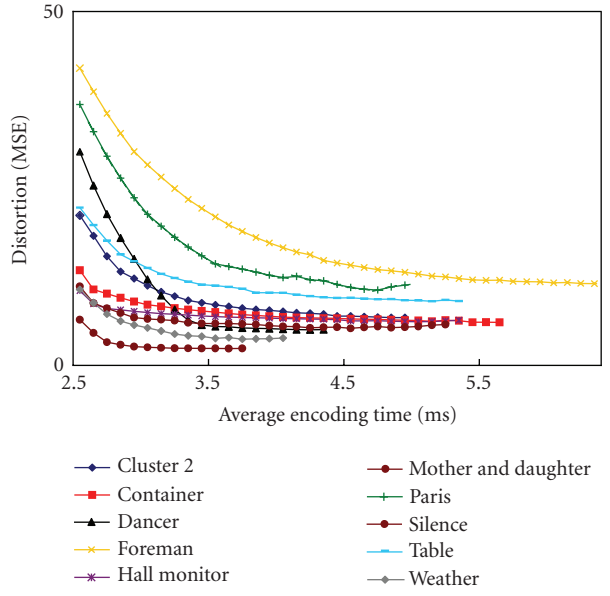


Figure 12: The relationship between average encoding time and distortion for video sequences in cluster 2.

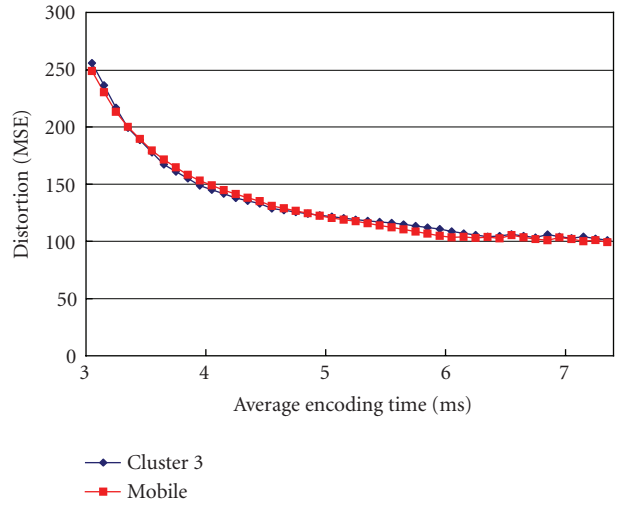


Figure 13: The relationship between average encoding time and distortion for video sequences in cluster 3.

T_{target}. To keep T_{average} close to T_{target}, T_D records the accumulation error of T_{actual}, determined by

$$T_{D,t} = T_{D,t-1} + (T_{\text{actual}} - T_{\text{target}}), \tag{9}$$

where $T_{D,t}$ denotes the current value of T_D and $T_{D,t-1}$ denotes the previous value of T_D. With the current value of T_D, the available encoding time for the subsequent frames is determined by

$$T_{\text{available}} = T_{\text{target}} - \alpha T_D, \quad 0 < \alpha < 1. \tag{10}$$

Table 4: Results of resource allocation with global optimization (GO) versus without GO for two high-priority and two low-priority channels.

Sequence	Complexity allocation mechanism	Target average encoding time for each frame (ms)	Actual average encoding time for each frame (ms)	Average PSNR for the 1st high-priority encoder (dB)	Average PSNR for the 1st low-priority encoder (dB)
Bus	Without GO	40	40.65	32.16	24.89
	With GO	40	40.01	31.95	31.02
	Without GO	50	50.02	33.36	32.09
	With GO	50	50.01	33.32	32.83
	Without GO	60	60.00	34.39	34.18
	With GO	60	60.00	34.38	34.26
Canoa	Without GO	40	40.32	30.91	25.15
	With GO	40	40.01	30.68	29.55
	Without GO	50	50.01	31.99	31.05
	With GO	50	50.00	31.94	31.61
	Without GO	60	60.01	32.94	32.82
	With GO	60	60.00	32.93	32.88
Coastguard	Without GO	40	40.10	34.59	28.87
	With GO	40	39.99	33.22	34.26
	Without GO	50	50.01	35.70	33.35
	With GO	50	50.00	35.61	34.69
	Without GO	60	60.02	36.66	36.02
	With GO	60	60.00	36.59	36.42
Football	Without GO	40	40.11	34.49	28.31
	With GO	40	40.01	34.32	33.19
	Without GO	50	50.02	35.70	35.16
	With GO	50	50.01	35.65	35.45
	Without GO	60	57.73	36.73	36.72
	With GO	60	57.66	36.74	36.72
Stefan	Without GO	40	40.51	32.39	24.86
	With GO	40	40.01	32.17	31.24
	Without GO	50	50.01	33.68	32.65
	With GO	50	50.01	33.62	33.10
	Without GO	60	60.00	34.79	34.69
	With GO	60	60.01	34.77	34.71

The goal of subtracting T_D from T_{target} is to reduce the accumulation error, and α determines the speed of complexity adjustment. When α approaches one, the accumulation error approaches zero more quickly but the fluctuation of the video quality is large. If α approaches zero, the accumulation error approaches zero slowly, but the fluctuation of the video quality is much smaller. Therefore, choosing the appropriate value of α makes the accumulation error stabilize more quickly and also smoothes the fluctuation of the video quality. In our simulations, value of α is set to 1/3, which makes the accumulation error always smaller than 40 milliseconds, equivalent to the interval of one single frame.

6. GLOBALLY OPTIMIZED RESOURCE ALLOCATION

In this section, a globally optimized resource allocation mechanism is developed to minimize the overall distortion rather than only those of high-priority channels. By the complexity-distortion model developed in Section 4, the ARC module is able to predict the encoding time and distortion of the subsequent frames for all combinations of M_{Total} and N_{NZMB}. The optimal resource allocation is obtained by selecting the values of M_{Total} and N_{NZMB} that minimize the predicted global distortion. The optimization problem is formulated as

$$\{\mathbf{p_i}\} = \arg\min_{\mathbf{p_i}} \sum_{i=1}^{N_H} D_i(\mathbf{p_i}) + \lambda \sum_{i=N_H+1}^{N_H+N_L} D_i(\mathbf{p_i}) \quad \text{s.t.} \sum_{i=1}^{N_H+N_L} C_i(\mathbf{p_i}) \le T_{\text{available}}, \quad 0 < \lambda < 1. \tag{11}$$

Equation (11) decides the encoding parameter vector $\mathbf{p_i}$ to minimize the sum of distortions while satisfying the constraint on $T_{\text{available}}$, determined in (10). The coefficient λ represents the weighting of the distortions of low-priority channels. If λ is set to 1, the distortions of low-priority channels are considered as important as those of high-priority channels. If λ is close to 0, the distortions of low-priority channels are not considered at all. The calculation of (11) involves searching among various $\mathbf{p_i}$, which is very

Table 5: Results of resource allocation with GO versus without GO for two high-priority and four low-priority channels.

Sequence	Complexity allocation mechanism	Target average encoding time for each frame (ms)	Actual average encoding time for each frame (ms)	Average PSNR for the 1st high-priority encoder (dB)	Average PSNR for the 1st low-priority encoder (dB)
Hall monitor	Without GO	45	46.32	40.83	37.91
	With GO	45	45.01	40.80	40.16
	Without GO	55	55.02	41.43	41.10
	With GO	55	55.02	41.43	41.26
	Without GO	65	65.00	41.91	41.87
	With GO	65	64.98	41.91	41.89

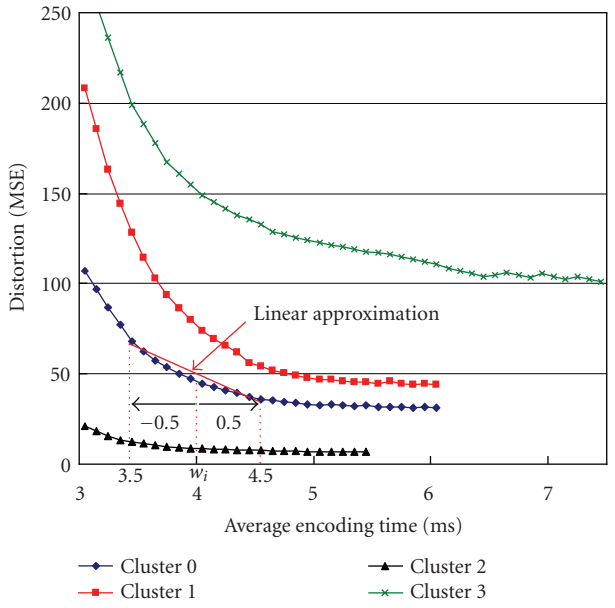


Figure 14: Piecewise linear approximation of the distortion function.

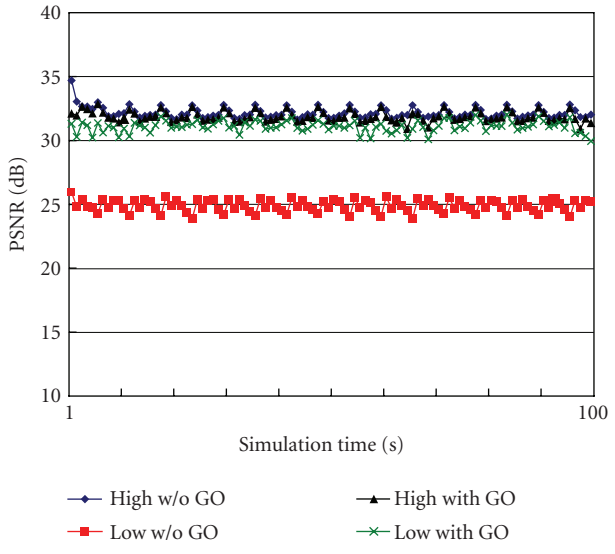


Figure 15: The PSNR values of the Bus sequence when the target encoding time for each frame is 40 milliseconds.

time-consuming. From Figure 14, the complexity-distortion functions are nonlinear, and this implies that it is not possible to use a simple linear function to replace the searching process to reduce the execution time. This execution overhead of resource allocation is intolerable for live streaming with a real-time constraint.

To reduce the execution time of the globally optimized resource allocation, the piecewise linear approximation technique is applied to simplify the optimization problem into a linear programming problem solvable by the simplex algorithm in real time. The procedure of linear approximation of the ith channel is shown in Figure 14. Assume that the system is processing a sequence belonging to cluster 0 and the current operation point w_i equals 4 milliseconds, the lower bound of the complexity control is set to $w_i - 0.5$ and upper bound is set to $w_i + 0.5$. The upper bound and lower bound must be within the working range of the complexity-distortion curve. Then, linear approximation is applied to the curve of cluster 0 within the range of complexity control. The linearly approximated version of the complexity-distortion function within this range is expressed as

$$D_i(x) = m_{w_i}x + d_{w_i}, \quad w_i - 0.5 < x < w_i + 0.5, \tag{12}$$

where m_{w_i} is the slope of the approximated segment, which is calculated by

$$\begin{aligned} m_{w_i} &= \frac{D_i(w_i + 0.5) - D_i(w_i - 0.5)}{(w_i + 0.5) - (w_i - 0.5)} \\ &= D_i(w_i + 0.5) - D_i(w_i - 0.5). \end{aligned} \tag{13}$$

Based on (12) and (13), the global optimization problem of (8) is rewritten as

$$\begin{aligned} \{\mathbf{p_i}\} = \arg\min_{\mathbf{p_i}} \Bigg(& \sum_{i=1}^{N_H} (m_{w_i} C_i(\mathbf{p_i}) + d_i) \\ & + \lambda \sum_{i=N_H+1}^{N_H+N_L} (m_{w_i} C_i(\mathbf{p_i}) + d_i) \Bigg) \\ \text{s.t.} \ & \sum_{i=1}^{N_H+N_L} C_i(\mathbf{p_i}) \le T_{\text{available}}, \\ & w_i - 0.5 \le C_i(\mathbf{p_i}) \le w_i + 0.5, \quad 0 < \lambda < 1. \end{aligned} \tag{14}$$

Equation (14) can be further simplified because the values of d_i are constants for a fixed w_i. Therefore, all terms of

d_i are removed, and the simplified version of (14) is written as

$$\{\mathbf{p_i}\} = \arg\min_{\mathbf{p_i}} \left(\sum_{i=1}^{N_H} (m_{w_i} C_i(\mathbf{p_i})) + \lambda \sum_{i=N_H+1}^{N_H+N_L} (m_{w_i} C_i(\mathbf{p_i})) \right)$$
$$\text{s.t.} \sum_{i=1}^{N_H+N_L} C_i(\mathbf{p_i}) \le T_{\text{available}}, \quad w_i - 0.5 \le C_i(\mathbf{p_i}) \le w_i + 0.5, \quad 0 < \lambda < 1. \tag{15}$$

Equation (15) represents a linear programming problem solvable by the simplex algorithm. The GNU linear programming kit (GLPK) package [22] is applied to solve the optimization problem, and the number of variables is equal to the number of channels. To ensure real-time performance, typical simulations are performed to test GLPK. Table 3 shows that the execution time to solve the linear programming problem with seven variables is only 0.335 milliseconds, which shows the real-time performance of the globally optimized allocation, as previously claimed.

7. SIMULATION RESULTS

The simulations are performed on a computer with Pentium 4 CPU 3.2 GHz and 768 MB RAM. In the simulations, we only use I and P frames with I frame interval being 30 frames, and the target bit rate for each channel is fixed at 1 Mbps. The value of α is set to 1/3, and the value of λ is set to 1/10. In the following figures and tables, the term "GO" is used to represent global optimization. For clarity, only one channel of each priority group is illustrated because the results of other channels are similar to the representatives shown.

There are three simulations. The first simulation consists of two high-priority and two low-priority channels, and the target encoding time for each frame is fixed to 40 milliseconds. The simulation results in Figure 15 show the comparison between the video quality of channels with global optimization and that of channels without global optimization. The video quality of the high-priority channel with global optimization is almost the same as the one without global optimization. However, the video quality of the low-priority channel with global optimization is much better than that without global optimization. The reason is that the allocation mechanism without global optimization allocates most of the computational resource to high-priority channels without considering the quality of low-priority channels. The globally optimized allocation mechanism allocates more computational resource to low-priority channels when it finds that the video quality of high-priority channels improves only a little even with more resource. The second simulation also consists of two high-priority and two low-priority channels. To test the accuracy of the proposed allocation scheme, different time constraints are set. The results are listed in Table 4. The minimum target average encoding time is set to 40 milliseconds because of limited computing power. These tables show that the video quality for two resource allocation schemes is the same when the target encoding time is high. However, when the target encoding time is low, the video quality of low-priority channels without global optimization is much lower than that with global optimization. Besides, the actual average encoding time for the globally optimized allocation mechanism is still very close to the target encoding time, but the allocation mechanism without global optimization becomes inaccurate when the target encoding time is 40 milliseconds.

The third simulation consists of two high-priority and four low-priority channels, and the results are shown in Table 5, which are similar to those in Table 4. The minimum target average encoding time here is set to 45 ms because there are more channels for the server to handle. The globally optimized resource allocation scheme performs better when the target encoding time is low. Based on the simulation results, we can conclude that the globally optimized resource allocation scheme does improve the quality of low-priority channels a lot with little quality drop of high-priority channels.

8. CONCLUSION

In summary, we have presented the design of a complexity-aware live streaming system, which utilizes the complexity-distortion model to allocate the computational resource to each channel in a global optimization way. To reduce the execution time of resource allocation, we formulate the optimization problem as a linear programming problem with piecewise linear approximation of the complexity-distortion model. In addition, a block-based complexity control method, which allows the system to accurately control the computational resource of each channel on the live streaming server, has also been developed. For sequences with varying encoding characteristics, our system is also able to find the optimal strategy by recalculating the parameter values of the C-D model because of the small computational overhead. The simulation results demonstrate the effectiveness of the proposed techniques.

ACKNOWLEDGMENTS

This work was supported in part by grants from the Intel Corporation and the National Science Council of Taiwan under Contracts NSC 94-2219-E-002-016, NSC 94-2219-E-002-012, and NSC 94-2725-E-002-006-PAE.

REFERENCES

[1] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 246–250, 1997.

[2] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1221–1236, 2001.

[3] Z. He and S. K. Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 970–982, 2002.

[4] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 172–185, 1999.

[5] P.-L. Tai, S.-Y. Huang, C.-T. Liu, and J.-S. Wang, "Computation-aware scheme for software-based block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 9, pp. 901–913, 2003.

[6] C.-Y. Chen, Y.-W. Huang, C.-L. Lee, and L.-G. Chen, "One-pass computation-aware motion estimation with adaptive search strategy," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 698–706, 2006.

[7] Y. Zhao and I. E. G. Richardson, "Complexity management for video encoders," in *Proceedings of the 10th ACM International Multimedia Conference*, pp. 647–649, Juan les Pins, France, December 2002.

[8] Z. Zhong and Y. Chen, "Complexity regulation for real-time video encoding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '02)*, vol. 1, pp. 737–740, Rochester, NY, USA, September 2002.

[9] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 5, pp. 645–658, 2005.

[10] J. Stottrup-Andersen, S. Forchhammer, and S. M. Aghito, "Rate-distortion-complexity optimization of fast motion estimation in H.264/MPEG-4 AVC," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 1, pp. 111–114, Singapore, October 2004.

[11] M. van der Schaar, D. Turaga, and V. Akella, "Rate-distortion-complexity adaptive video compression and streaming," in *Proceedings of IEEE International Conference on Image Processing (ICIP '04)*, vol. 3, pp. 2051–2054, Singapore, October 2004.

[12] M. van der Schaar, Y. Andreonoulos, and O. Li, "Real-time ubiquitous multimedia streaming using rate-distortion-complexity models," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 2, pp. 639–643, Dallas, Tex, USA, November-December 2004.

[13] G. Landge, M. van der Schaar, and V. Akella, "Complexity metric driven energy optimization framework for implementing MPEG-21 scalable video decoders," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 2, pp. 1141–1144, Philadelphia, Pa, USA, March 2005.

[14] M. van der Schaar and Y. Andreopoulos, "Rate-distortion-complexity modeling for network and receiver aware adaptation," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 471–479, 2005.

[15] S.-F. Lin, M.-T. Lu, H. H. Chen, and C.-H. Pan, "Fast multiframe motion estimation for H.264 and its applications to complexity-aware streaming," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '05)*, vol. 2, pp. 1505–1508, Kobe, Japan, May 2005.

[16] M.-T. Lu, C.-K. Lin, J. J. Yao, and H. H. Chen, "Complexity-aware live streaming system," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 193–196, Genova, Italy, September 2005.

[17] M.-T. Lu, C.-K. Lin, J. J. Yao, and H. H. Chen, "A complexity-aware live streaming system with bit rate adjustment," in *Proceedings of the 7th IEEE International Symposium on Multimedia (ISM '05)*, pp. 431–437, Irvine, Calif, USA, December 2005.

[18] M.-T. Lu, C.-K. Lin, J. J. Yao, and H. H. Chen, "Block-based computation adjustment for complexity-aware live streaming systems," in *Proceedings of the Picture Coding Symposium*, Beijing, China, April 2006.

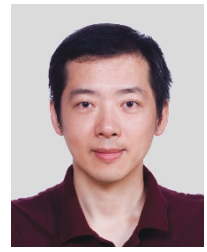
[19] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a Transport Protocol for Real-Time Applications," *Request for Comments 3550*, IETF Network Working Group, July 2003.

[20] Efficient Algorithms for K-Means Clustering, http://www.cs.umd.edu/~mount/Projects/KMeans/.

[21] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[22] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "A local search approximation algorithm for k-means clustering," in *Proceedings of the 18th Annual Symposium on Computational Geometry (SCG '02)*, pp. 10–18, Barcelona, Spain, June 2002.

Meng-Ting Lu was born in Peng-Hu, Taiwan. He received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, in 2003. He is currently working towards the Ph.D. degree in the Graduate Institute of Communication Engineering, National Taiwan University. His research interests include video streaming, peer-to-peer streaming, and video coding.



Jason J. Yao received his B.S. degree in electrical engineering from National Taiwan University, Taiwan, and his Ph.D. degree in electrical and computer engineering from University of California, Santa Barbara. His research interests span from digital signal processing, telecommunications, audio/video systems to Internet traffic engineering and bioinformatics. He has worked for AT&T Bell Labs, Fujitsu Network Transport Systems, Fujitsu Laboratories of America with job functions in advanced research, project management, and strategic planning. Dr. Yao also holds an MBA from Santa Clara University.

Homer H. Chen received the Ph.D. degree from University of Illinois at Urbana-Champaign in electrical and computer engineering. Since August 2003, he has been with the College of Electrical Engineering and Computer Science, National Taiwan University, where he is Irving T. Ho Chair Professor. Prior to that, he had held various R&D management and engineering positions in leading US companies including AT&T Bell Labs, Rockwell Science Center, iVast, and Digital Island over a period of 17 years. He was a US Delegate of the ISO and ITU standards committees and contributed to the development of many new interactive multimedia technologies that are now part of the MPEG-4 and JPEG-2000 standards. His research interests lie in the broad area of multimedia processing and communications. He is an IEEE Fellow.