Joint Video Coding and Statistical Multiplexing for Broadcasting Over DVB-H Channels

Mehdi Rezaei, *Member, IEEE*, Imed Bouazizi, *Member, IEEE*, and Moncef Gabbouj, *Senior Member, IEEE*

***Abstract*—A novel joint video encoding and statistical multiplexing (StatMux) method for broadcasting over digital video broadcasting for handhelds (DVB-H) channels is proposed to improve the quality of encoded video and to decrease the end-to-end delay in a broadcast system. The main parts of end-to-end delay in a DVB-H system result from a time-sliced transmission scheme that is used in DVB-H and from the bit rate variations of service bit streams. The time-sliced transmission scheme is utilized in DVB-H to reduce the power consumption of DVB-H receivers. Variable bit rate (VBR) video bit streams are used in DVB-H to improve the video quality and compression performance. The time-sliced transmission scheme has increased the channel switching delay, i.e., switching to a new audio-visual service, in DVB-H. The used VBR bit streams increase the required buffering delays in the whole system. The different parts of end-to-end delay in a DVB-H system can be affected by the used video encoding and multiplexing methods. Different scenarios for encoding and StatMux of video sources for DVB-H application are studied in this paper. Moreover, a new method for jointly encoding and StatMux of video sources is proposed that not only decreases the end-to-end delay but also improves the average quality of compressed video by dynamically distributing available bandwidth between the video sources according to their relative complexity. Performance of the proposed method is validated by simulation results.**

***Index Terms*—Channel switching, delay, DVB-H (digital video broadcasting-handhelds), fuzzy control, joint rate control, mobile TV, statistical multiplexing, video coding.**

I. INTRODUCTION

DIGITAL video broadcasting for handheld terminals (DVB-H) is an ETSI specification for delivering broadcast services to battery-powered handheld receivers [1], [2]. DVB-H is mainly based on the DVB-T specification for digital terrestrial television. However, it adds a number of features designed to consider the limited battery life of handheld devices and the particular environments in which such receivers operate.

A simplified block diagram of a conventional *IP Datacasting* (IPDC) system over DVB-H is depicted in Fig. 1. As shown, a content encoder receives a source signal and encodes the source signal into a coded media bit stream. Content encoder is typically capable of encoding more than one media type, such as audio and video. Alternatively, more than one content encoder may be required to code different media types of the source signal. Fig. 1 illustrates the processing of one coded media bit stream of one media type.

The coded media bit stream is transferred to a server. Content encoder and server may reside on the same physical device or may be included in separate devices. Content encoder and server may operate with live real-time content, in which case the coded media bit stream may not be stored permanently, but rather buffered for small periods of time in content encoder and/or in server to smooth out variations in processing delay, transfer delay, and coded media bit rate. Content encoder may also operate considerably earlier than when the bit stream is transmitted from the server. In such a case, the system may include a content database, which may reside on a separate device or on the same device as content encoder and/or server.

The server may be an IP multicast server using real-time media transport over *Real-time Transport Protocol* (RTP). The server is configured to encapsulate the coded media bit stream into RTP packets according to an RTP payload format. Although not shown in this figure, the system may contain more than one server. The server is connected to an *IP encapsulator*, also referred to as a *multi-protocol encapsulator*.

The IP encapsulator encapsulates IP packets into *Multi-Protocol Encapsulation (MPE) Sections* which are further packetized into MPEG-2 *Transport Stream* packets. The IP encapsulator optionally uses *MPE Forward Error Correction* (MPE-FEC) based on *Reed-Solomon* codes. An IPDC system over DVB-H further includes at least one radio transmitter which is not discussed further.

Services used in mobile handheld terminals require relatively low bit rates. The estimated maximum bit rate for streaming video using advanced compression technology such as H.264/AVC is in the order of a few hundred kilobits per second. A DVB-T transmission system usually provides a bit rate of 10 Mbps or more. This provides a possibility to significantly reduce the average power consumption of a DVB-H receiver by introducing a scheme based on time division multiplexing. This scheme is called *Time-slicing*. To reduce the power consumption in handheld terminals, the service data is time-sliced and then sent over the channel as bursts at a significantly higher bit rate compared to the bit rate of the audio-visual service. Time-slicing enables a receiver to stay active only during a small fraction of the time, while receiving bursts of a requested service. It significantly reduces the power consumption used for radio reception. A number of time-sliced services can be multiplexed by IP Encapsulator to fill a DVB-T

Manuscript received January 22, 2007; revised March 24, 2008. Current version published December 10, 2008. This work was supported in part by Nokia and the Academy of Finland under Project 213462 (Finnish Centre of Excellence program 2006—2011). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yo-Sung Ho.

M. Rezaei and M. Gabbouj are with the Department of Signal Processing, Tampere University of Technology, Finland, FI-33720 (e-mail: mehdi.rezaei@ieee.org; Moncef.Gabbouj@tut.fi).

I. Bouazizi is with the Nokia Research Center, Tampere, Finland, FI-33720 (e-mail: Imed.Bouazizi@nokia.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2008.2007315

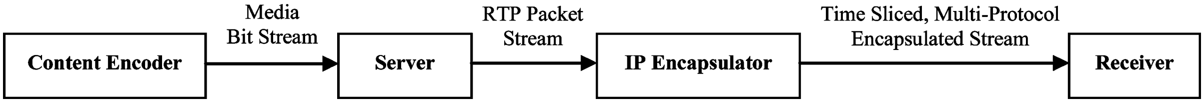

Fig. 1. Simplified block diagram of a conventional IP datacasting system.

transmission channel. DVB-H also employs additional forward error correction to further improve mobile and indoor reception performance DVB-T.

The end-to-end delay of a broadcast system over DVB-H includes several parts. A delay is required for initial buffering of service data in the IP Encapsulator. This delay can be divided into two parts. First, a buffering delay proportional to the burst size is required for receiving enough data for encapsulating the first MPE-FEC frame. This delay is called *Encapsulation Buffering Delay* in this paper. Second, a buffering delay, namely *Pre-encapsulation Buffering Delay*, is required to compensate the variations that exist in the bit rates of service bit streams. Tune-in time is another part of end-to-end delay that is required when a receiver switches to a new service. The channel switching delay or tune-in time in DVB-H refers to the time between the start of switching and the start of the media rendering of the new channel. Tune-in time can be divided into several parts [3]. The main parts of tune-in time are: *Arrival Delay* (delay to arrival of desired burst), *Reception Delay* (reception duration of desired burst), *Decapsulation Buffering Delay*, *Decoder Refresh Delay* (delay to the first random access point e.g., an *Instantaneous Decoder Refresh* (IDR) picture in AVC\H.264 video coding standard), *Decoder Buffering Delay* (Initial buffering period of *Coded Picture Buffer*). The decapsulation buffering delay includes two buffering delays for the *Multi-protocol Decapsulation Buffer* (MDB) and *RTP (Real Time Protocol) Decapsulation Buffer* (RDB) [4], [5]. The decapsulation buffering delay is required to compensate the variations of burst size and the decoder buffering delay is needed to compensate the variations that exist in the bit rates of bit streams. Moreover, another delay is needed for synchronization between the associated streams (e.g., audio and video) of the streaming session which is not shown in the graph. The end-to-end delay also includes other parts such as delay in an IP network that are not affected by the proposed video encoding and multiplexing methods a great deal and hence they are not discussed further.

Considering the sources of delays, a new grouping on the different parts of end-to-end delay is possible. The first group includes the pre-encapsulation buffering delay, the RTP decapsulation buffering delay, and the decoder buffering delay. The overall delay of this group, called *Rate-Dependent Buffering Delay*, depends on to the variations in the bit rates of bit streams and thereafter depends on the encoding and rate control parameters. For video streaming over DVB-H application the advantages of variable bit rate video is exploited. For most video content, a variable bit rate video can provide better visual quality and coding efficiency than a constant bit rate video [6]. A higher quality and compression performance can be obtained by more variations in bit rate and higher rate-dependent buffering delay. The second group includes the encapsulation buffering delay, the arrival delay, the reception delay, and the MDB delay. The overall delay of this group, called *Time-Slicing Delay*, depends on the time-slicing parameters that define the power saving consumption of DVB-H receiver. The lower the receiver power consumption, the higher is the time-slicing delay. The third group includes the decoder refresh delay, the associated streams synchronization delay, the delay in IP network, and the remained parts of end-to-end delay. This group of delay is assumed that is not affected considerably by the rate control and time-slicing parameters.

A joint video and StatMux method is proposed in this paper that can decrease the end-to-end delay of a DVB-H broadcast system and it can also improve the overall quality of compressed video. The major problem of joint video encoding and StatMux is how to allocate available bit budget among the video sources that share a common channel bandwidth and are jointly encoded. This bit allocation should be based on the instantaneous coding complexity of each video source and also based on the relative complexities of the video sources. Existing methods in the literature for this scenario follow two main approaches: *Forward Analysis* and *Modeling Strategy*. In forward analysis a preprocessing is performed on video sources to gather statistics about the coding complexity and scene changes. The real coding process can operate based on the statistics obtained by the pre-analysis. In the modeling approach, first it is attempted to model the performance of video encoder and the coding complexity of video sources and then the allocated bits to the video sources are controlled based on the provided models, while the models parameters are updated during encoding. See the proposed methods in [7]–[10] as examples for the two approaches. The technique presented in [7] is a joint encoding method for a multi-program broadcasting of pre-encoded video. The authors try to distribute the available bandwidth between as many programs as possible with a level of quality according to the complexity of each program. They first encode all the programs with a high bit rate and quality while all relevant statistics on the complexities of the video streams are collected. The encoded video along with the gathered statistics are stored in the video server. When streaming, a joint rate controller uses the statistics to perform the bit allocation to the bit streams which are transcoded by a set of transcoders. In [8], a joint rate control algorithm for a multiprogram video compression system is presented. The proposed joint rate control operates based on feedback and look-ahead approaches. The system consists of several preprocessors and video encoders. Each preprocessor analyzes a video source and derives picture statistics characterizing the scene content of that video source. The coding statistics generated by the preprocessors and the encoders are input parameters to the joint rate controller. Using these statistics, the joint rate controller calculates dynamically the bit rate for each encoder based on the relative complexities of the programs and occurrence of scene cuts in the programs. According to the proposed joint rate control algorithm, each encoder changes its bit rate at

GOP (group of picture) boundaries or at scene cuts. Another bit allocation method for joint coding of multiple video programs is presented in [9]. In this method, the input video programs are divided into *Super GOPs* that are identical in terms of the number of frames of each type (I, P, and B). The authors also define a *Super Frame* as a collection of frames, one from each of the programs taken at the same time instant. Then, it is tried to distribute the bit budget hierarchically between the video super GOPs, super frames, and frames according to their relative complexities while the encoder and decoders' buffers are prevented from overflowing and under flowing. Finally, using a rate-distortion model, a quantization parameter (QP) is calculated for each frame according to the allocated bits to the frame. A similar method is presented in [10]. In this paper, it is tried to distribute the bit budget between the super GOPs, super frames, frames and also macroblocks according to their relative complexities.

The proposed joint rate control algorithm in this paper is optimized for a DVB-H broadcast system to provide a near constant and common quality for the encoded video bit streams and to decrease the rate-dependent buffering delay as a major part of end-to-end delay. The proposed method in this paper is different from the previous studied methods from several points of view. First, it is a real-time method without any look ahead even for one frame. Second, unlike the previous methods, it does not use any bit allocation strategy. The bit rates of bit streams are controlled by QP on a picture basis while the controller does not compute the QP directly. Utilizing a fuzzy controller, the controller only computes the variations of QP. Furthermore, thanks to the fuzzy controller, the proposed method has a very low degree of complexity in comparison to the studied methods. Moreover, it has a granular structure such that it can be used for encoding any number of bit streams. Finally, it is tuned easily to achieve the desired bit rate, quality, and delay.

The rest of this paper is organized as follows. In Section II, a study on how different scenarios of video encoding and multiplexing methods that can affect the overall performance of a broadcast system in terms of end-to-end delay, video quality and the receiver power consumption is presented. Section III, evaluates the performance of StatMux regarding the reduction of end-to-end delay independently of the implementation method. The details of proposed joint video encoding and StatMux method are presented in Section IV. Simulation results are provided in Section V. Conclusions are presented in Section VI.

II. STATISTICAL MULTIPLEXING IN DVB-H

In video broadcasting over DVB-H, the video sources are encoded in variable bit rate (VBR) bit streams to utilize the advantage of VBR. When VBR bit streams are used, utilizing statistical multiplexing would be beneficial. In StatMux, a fixed bandwidth communication channel is divided into several variable bandwidth channels. The resulting VBR channels are adapted to the instantaneous traffic demands of the bit streams that are transferred over the channels. StatMux is used in many communication applications to improve the overall performance of communication channels. The performance of a communication channel can be evaluated in terms of transmission delay, data drop rate, and bandwidth utilization. When in DVB-H broadcast system, the bit rates of audio-visual services are controlled during encoding, achieving a high level of bandwidth utilization and a low level of data drop rate is not very difficult even without StatMux. However, DVB-H broadcast system can utilize StatMux to improve the overall quality of broadcast services and to decrease the system end-to-end delay. In a DVB-H broadcast system, StatMux can be implemented on VBR bit streams in the encoder and\or in the IP encapsulator.

When a number of video sources are encoded and broadcasted simultaneously and the video encoders are collocated geographically, a form of StatMux can be implemented in the encoders and in conjunction with encoding. Statistical multiplexing in the encoders is implemented by a joint rate controller that simultaneously controls the encoders. The joint rate controller distributes the available bandwidth among the video sources according to their coding complexities to provide a higher overall quality for compressed video. Moreover, the variations in the bit rate of an aggregated bit stream can be controlled by the joint rate controller to decrease the overall required rate-dependent buffering delay of a broadcast system.

StatMux in an IP encapsulator can be implemented in the time domain and in conjunction with time-slicing. During time-slicing a *time division multiplexing* (TDM) is implemented by an IP encapsulator on a number of time-sliced services to fill a DVB-T transmission channel. The TDM can be deterministic or statistical. StatMux in an IP encapsulator means a statistical TDM in which the lengths of time-slices are proportional to the instantaneous bandwidth demands of bit streams. Statistical TDM decreases the overall rate-dependent buffering delay of a DVB-H system by allocating variable bandwidth channels to the bit streams proportional to variations in their bit rates. In an ideal case, the pre-encapsulation buffering delay, the decoder buffering delay and the RTP buffering delay are minimized to zero.

Another form of StatMux can be implemented by an IP encapsulator inside the time-slices. As an option in DVB-H, a number of services can be encapsulated to one time-slice or burst [11]. In this case the available bandwidth for the time-slice can be statistically distributed among the services. The resulting time-slices by this intra-slice multiplexing process can be multiplexed again as a deterministic or statistical TDM. The intra-slice multiplexing provides a new degree of freedom for StatMux in DVB-H.

When a number of services are multiplexed to one time-slice that is completely received by the receiver, not only the rate-dependent buffering delay decreases but also the time-slicing delay will decrease if the receiver switches among the services that are inside the same time-slice. In this case, the arrival delay and the reception delay are minimized to zero. Moreover, considering a uniform probability distribution function for the channel switching time, on the average it is expected to have a reduction of about 50% in the decapsulation delay and in the decoder refresh delay. The reduction of time-slicing delay is achieved at the expense of an increase in power consumption of a DVB-H receiver. The increase in power consumption depends on the size of the time-slice and it can be reasonably low when few services are carried by the time-slice.

In practice, a combination of joint encoding and independent encoding, deterministic TDM and statistical TDM, inter-slice, and intra-slice multiplexing can be used in a DVB-H broadcast system. For example, an aggregated bit stream resulting from a joint encoding system can be targeted to the intra-slice multiplexing to be easily encapsulated to one time-slice. In this case, the time-slicing delay decreases at the expense of an increase in the receiver power consumption. As another option, the aggregated bit stream can be targeted to the statistical TDM and to be distributed between multiple time-slices. In this case, the rate-dependent buffering delay decreases independently of time-slicing delay and the receiver power consumption. While the encoding method only affects the rate-dependent buffering delay, the power consumption of a DVB-H receiver is almost independent of the encoding method. The TDM can affect one or all three groups of delays including rate-dependent buffering delay, time-slicing delay, and common delay. The implementation of TDM in an IP encapsulator is out of scope of this paper.

The rate-dependent buffering delay as a major part of end-to-end delay of DVB-H broadcast system can be reduced by StatMux in the encoder or/and in an IP encapsulator. The performance of StatMux in terms of delay reduction depends on the statistical properties of encoded bit streams. It also depends on the number of services that are multiplexed. Before implementing any StatMux in an encoder or an IP encapsulator, we need to roughly evaluate the possible performance that can be achieved by StatMux of DVB-H services. In Section III, we try to evaluate the performance of StatMux on DVB-H services independently of the underlying implementation technique.

III. PRELIMINARY EVALUATION OF STATISTICAL MULTIPLEXING PERFORMANCE ON DVB-H SERVICES

Generally, a StatMux algorithm operates in a 3-D space including bandwidth utilization, data drop rate, and delay dimensions. To roughly evaluate the performance of StatMux, a simple operation point is considered in which the maximum bandwidth is utilized and the data drop rate is zero. Therefore, the performance of StatMux can be simply compared with deterministic multiplexing in terms of delay reduction.

Consider a number of N bit streams that are encoded independently with the average bit rates of $r_1, r_2, \ldots, r_N$, respectively. The nth video bit stream has a number of M pictures with sizes $b_n^1, b_n^2, \ldots, b_n^M$ bits. A constant bandwidth channel is going to be used for transmission of these bit streams. Using the whole bandwidth, R would then be

$$R = \sum_{i=1}^{N} r_i \tag{1}$$

In ideal StatMux, the bandwidth is shared between the bit streams proportionally to their instantaneous bit rates and relative complexities. In other words, a VBR channel is allocated to each bit stream. The allocated bandwidth to the nth bit stream, when transmitting the mth picture, is given by

$$r_n^m = \frac{b_n^m R}{\sum\limits_{i=1}^{N} b_i^m} \tag{2}$$

where r_n^m denotes the temporal bandwidth allocated to the nth bit stream when the mth picture is transmitted. Bypassing the encapsulation and decapsulation process, the transmission time of the mth picture in the nth bit stream is calculated as

$$T_n^m = \frac{b_n^m}{r_n^m} = \frac{1}{R}\sum_{i=1}^{N} b_i^m \tag{3}$$

and it yields

$$T_1^m = T_2^m = \cdots = T_N^m. \tag{4}$$

From an implementation point of view, this means that ideal StatMux is equivalent to synchronous transmission of corresponding pictures in the video bit streams. If the multiplexed services are encapsulated into one burst, ideal StatMux means encapsulating similar numbers of video pictures from different bit streams into one burst. If the multiplexed services are encapsulated in to separate bursts, StatMux means having similar number of video pictures in the bursts.

TABLE I
RESULTS OF STATISTICAL MULTIPLEXING ON AVERAGE BUFFERING DELAY

Video Sequence Sets	**Average Buffering Delay (s)**		**Delay Reduction%**
	DetMux	***StatMux***	
$_{4}C_{2}$	0.48	0.41	15
$_{4}C_{3}$	0.49	0.37	24
$_{5}C_{4}$	0.50	0.34	32
$_{6}C_{5}$	0.48	0.31	36
$_{7}C_{6}$	0.47	0.27	43

To preliminarily evaluate the performance of StatMux on DVB-H services from a delay reduction point of view, a simulation was run on a set of typical video sequences. A number of seven video sequences (QVGA picture format, 15 f/s, 900 frames) with different contents including (Sport, News, Movie and Music) captured from TV are encoded independently. The bit streams are encoded for a target bit rate of 300 kb/s by the Nokia H.264/AVC codec that is available for the public [12]. The VBR rate control algorithm proposed for a DVB-H application in [13] was used with a buffer size of 300 kb. The described StatMux above, transmission, and reception were simulated on all combinations $_{n}\mathrm{C}_{r}$ (n =4, 5, 6, 7 and r =2, 3, 4, 5, 6) of video bit streams. The averages of required decoder buffering delays for the zero drop rate case over all subsets were measured and compared with the case in which the bit streams are transmitted independently and with deterministic multiplexing (DetMux). Simulation results are presented in Table I. Comparing the results of the two cases shows that StatMux can decrease the buffering delay about 15% if only two bit streams are multiplexed. When the number of multiplexed bit streams increases to 6, the required buffering delay can be reduced up to 43%. The more bit streams are multiplexed, the higher the reduction in buffering delay is provided.

When a DVB-T transmission channel is shared between DVB-H services, roughly 15–20 DVB-H services can be multiplexed and allocated in the channel. For this number of services, according to the provided simulation results, a

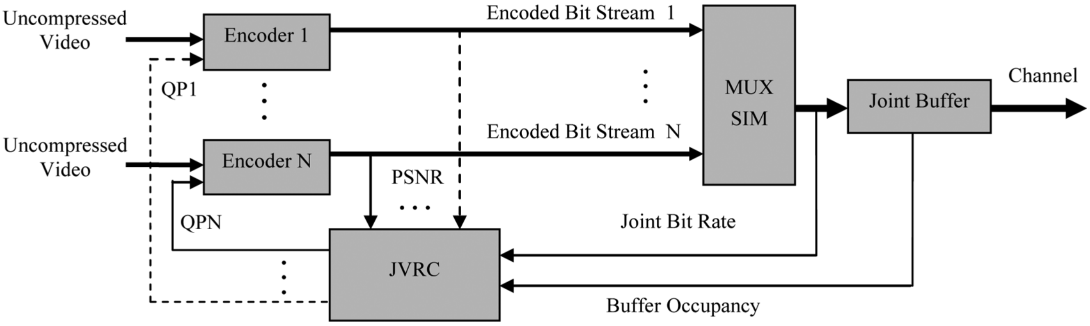

Fig. 2. Block diagram of proposed video encoding method.

considerable performance for StatMux in DVB-H is expected. In the Section IV, a novel fuzzy joint rate controller is proposed that can be used for implementation of StatMux in conjunction with video encoding.

IV. PROPOSED JOINT VIDEO ENCODING AND STATISTICAL MULTIPLEXING

According to the proposed method, StatMux is implemented in conjunction with the encoding process by a joint rate control algorithm. The joint rate control algorithm provides a near constant (along the bit streams) and common (across the bit streams) quality for the encoded video bit streams by dynamic distribution of the available bandwidth between the bit streams according to their relative complexities and also by minimizing the variations of QP. Furthermore, it controls the variations in the bit rate of the aggregated bit stream to decrease the rate-dependent part of the end-to-end delay of a broadcast system.

A simple block diagram of the proposed video encoding method is depicted in Fig. 2. As shown, a number of video sources are encoded simultaneously to bit streams by separate encoders. A joint video rate controller (JVRC) controls the bit rate and the quality of encoded bit streams by computing a quantization parameter on a picture basis for each encoder. The JVRC utilizes a virtual joint buffer and a multiplex simulator (MUX SIM). The multiplex simulator simulates the ideal case of StatMux which is implemented by the IP encapsulator. With this structure in which no feedback from real multiplexing is used, the bit streams can be encoded independently of real multiplexing and without any feedback from the IP encapsulator. The joint buffer is charged by the multiplexed bit streams and drained by a constant bit rate equal to the channel bandwidth. However, it is possible to consider that the virtual buffer is running on the receiver side. In that case, it is charged by a constant bit rate equal to bandwidth and it is discharged by the multiplexed bit streams. The JVRC uses a feedback signal from the occupancy of joint buffer and a feedback from the bit rate of an aggregated bit stream to control the overall bit rate. Also, it uses other feedback signals from the distortion (PSNR) of encoded bit streams to distribute the available bandwidth between the bit streams proportional to their coding complexity. The JVRC can be functionally divided into two separate controllers: a *Fuzzy Rate Controller* and a *Quality Controller*. An internal block diagram for JVRC is depicted in Fig. 3. More details about the joint encoding system are presented in the sequel.

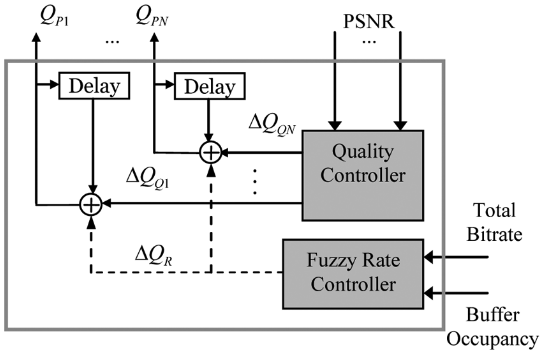

Fig. 3. Internal diagram of JVRC.

A. Virtual Joint Buffer

Considering the joint buffer at the receiver side, the occupancy of joint buffer is updated after encoding a series of corresponding pictures called *Super Picture*, one picture from each video source, as:

$$O(m) = O(m-1) - \sum_{i=1}^{N} b_i^m + \frac{R}{F} \tag{5}$$

where O denotes the occupancy of joint buffer and m shows the index of super picture. F stands for the frame rate and R indicates the target bit rate for the aggregated bit stream. Note that according to explanation of (4), the functionality of multiplex simulator is hidden in the buffering model (5) in which the video pictures are moved synchronously.

B. Joint Video Rate Controller

The JVRC computes a QP for each encoder based on the output of fuzzy controller (ΔQ_R), the corresponding output of quality controller (ΔQ_{Qn}) and the QP used for encoding previous picture in the encoder. The QP for each encoder is computed as

$$Q_n(i) = Q_n(i-1) + [\Delta Q_R + \Delta Q_{Qn}] \tag{6}$$

where n and i denote the indexes of encoder and picture, respectively. $[y]$ stands for the integer part of y. The output of fuzzy controller and quality controller are small values around zero which is added to the QP used for encoding previous picture. In this structure, only the variations of QP are computed. The Fuzzy controller has been optimized to minimize the variations of QP and thereafter to provide a more constant quality for the bit streams. The quality controller tries to balance the quality across the bit stream. More details about the fuzzy rate controller and the quality controller are presented in the sequel.

TABLE II
SUMMARIZATION OF THE IF-THEN FUZZY RULES

x_2

VH	6VH	5VH	4VH	3VH	2VH	VH	H	MH	M
H	5VH	4VH	**3VH**	2VH	VH	H	MH	M	ML
MH	4VH	3VH	2VH	VH	H	MH	M	ML	L
M	3VH	2VH	VH	H	MH	M	ML	L	VL
ML	2VH	VH	H	MH	M	ML	L	VL	2VL
L	VH	H	MH	M	ML	L	VL	2VL	3VL
VL	H	MH	M	ML	L	VL	2VL	3VL	4VL
	3VL	**2VL**	**VL**	**L**	**ML**	**M**	**MH**	**H**	**VH**

x_1

C. Fuzzy Rate Controller

Generally, fuzzy controllers can be designed based on experimental results and\or human expertise. In design of proposed fuzzy controller we used provided experimental results and expertise from our previous video rate controllers proposed for other applications [13]–[15]. Any analytical and empirical result can be used in this approach. The fuzzy controller is selected for this structure because, the nonlinearity and the complexity that exist between the elements of system can be simply included in the fuzzy rules and fuzzy membership functions. Moreover, according to the block diagram shown in Fig. 3, a controller is needed to determine a small quantized value based on rough measurements on the bit rate and the buffer fullness. These properties make it fit to a fuzzy controller with low resolution inputs and output.

The fuzzy controller has been designed such that to suppress the fluctuation of QP as much as possible while the buffer constraint is obeyed. Generally, the average perceived visual quality is maximized, when the video pictures are encoded with as constant quality as possible [3]. Furthermore, for a uniform video signal (such as a video scene), experimental and analytical results show that minimizing the variation in QP provides the almost maximum qualitative measure of quality [16].

The fuzzy controller has two input signals that are computed based on desired encoding parameters and also based on the received feedback signals from the buffer occupancy and the bit rate of aggregated bit stream. The inputs of fuzzy system are defined as

$$x_1 = \frac{O}{S} \tag{7}$$

$$x_2 = \frac{I_I + X_{IP} - 1}{I_I} \cdot \frac{F}{R}\left(\sum_{i=1}^{N_P} b_{Pi} + \frac{1}{X_{IP}}\sum_{i=1}^{N_I} b_{Ii}\right) \tag{8}$$

where x_1 and x_2 correspond to the inputs of fuzzy system. O and S stand for the occupancy and the size of joint buffer, respectively. b_{Pi} and b_{Ii} denote the consumed bit budgets by the ith P and I-picture respectively in the last encoded super picture. N_p and N_1 are the number of P and I-pictures in the last encoded super picture respectively and i shows the index of picture in the super picture. I_I indicates the interval of periodic I-pictures in the bit streams. F stands for the target frame rate. X_{IP} shows the average relative complexity of I-pictures to P-pictures and it is defined as

$$X_{IP} = \frac{\bar{b}_I}{\bar{b}_P} \tag{9}$$

where $\bar{b}_P$ and $\bar{b}_I$ denote the consumed bit budgets by the encoded P and I-pictures respectively in average over all encoded pictures. The input x_1 is a representation of buffer conditions. When the buffer is empty, $x_1 = 0$ and when the buffer is full, $x_1 = 1$. The input x_2 is the ratio of instantaneous bit rate to the target bit rate. The instantaneous bit rate is normalized according to the relative complexity and the combination of P and I-pictures in the super picture to ignore the variations in the bit rate result of high quality I-pictures.

The functionality between the inputs and the output of fuzzy controller is described by a number of fuzzy rules. An example fuzzy rule is:

> *If the buffer occupancy is close to the middle of buffer size and the current bit rate is close to the target bit rate, then do not change QP.*

This fuzzy rule can be denoted as

$$\begin{aligned} \text{IF} \quad & (x_1 \text{ is Medium}) \text{ AND } (x_2 \text{ is Medium}) \\ \text{THEN} \quad & (\text{Output is Medium}). \end{aligned}$$

The used fuzzy rules in the rate controller are summarized in Table II. The content of the table specifies the output of the controller according to the inputs. The letters H, L, M, and V correspond to fuzzy descriptions High, Low, Medium, and Very, respectively. The number before V shows the number of repetition or the level of strength. As an example from the table, it can be expressed as

$$\text{IF } (x_1 \text{ is VL and } x_2 \text{ is H}) \text{ THEN } (\text{Output is } 3\,\text{VH})$$

where 3 VH stands for Very Very Very High. The output of this rule is bold in the table.

The input signals are specified by their fuzzy membership functions (MSF). Seven and nine trapezoidal MSFs for the two inputs x_1 and x_2 were employed respectively. Each MSF can be

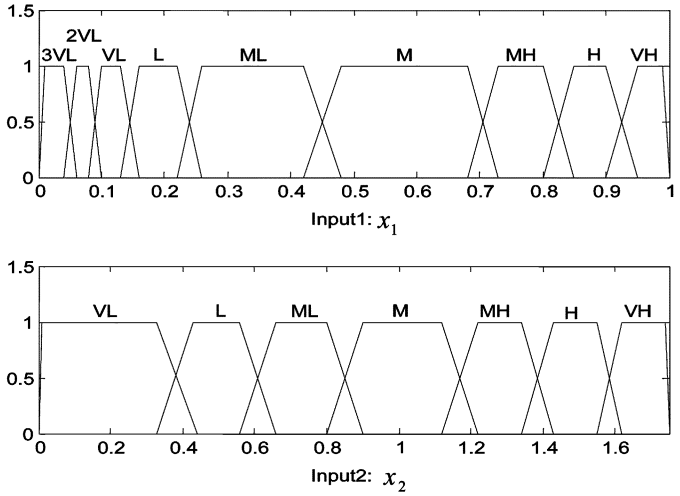

Fig. 4. Membership functions of the linguistic variables.

defined by the trapezoid corners. The whole MSFs are summarized in two matrices corresponding to the two inputs as shown below:

$$MSF_{x_1} = \begin{bmatrix} 0 & .01 & .08 & .12 \\ .08 & .12 & .16 & .20 \\ .16 & .20 & .26 & .30 \\ .26 & .30 & .38 & .42 \\ .38 & .42 & .52 & .56 \\ .52 & .56 & .68 & .72 \\ .68 & .72 & .82 & .85 \\ .82 & .85 & .92 & .95 \\ .92 & .95 & .99 & 1.0 \end{bmatrix}$$

$$MSF_{x_2} = \begin{bmatrix} 0 & .01 & .35 & .45 \\ .35 & .45 & .55 & .65 \\ .55 & .65 & .75 & .85 \\ .75 & .85 & 1.15 & 1.25 \\ 1.15 & 1.25 & 1.40 & 1.50 \\ 1.40 & 1.50 & 1.65 & 1.75 \\ 1.65 & 1.75 & 1.99 & 2.0 \end{bmatrix}.$$

The linguistic fuzzy rules and MSFs were designed based on some analytical results and provided experiences in [13]–[15]. Moreover, a heuristic optimization was run for the fine tuning of MSFs. The final distributions of MSFs are shown in Fig. 4. The desired central values for the output of fuzzy system correspond to Table II are presented in Table III.

A well-known fuzzy system with two inputs using *Product Inference Engine*, *Singleton Fuzzifier* and *Center Average Defuzzifier* was used:

$$f(x_1,x_2) = \frac{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \bar{y}^{i_1 i_2}\mu_{A_1^{i_1}}(x_1).\mu_{A_2^{i_2}}(x_2)}{\sum_{i_1=1}^{N_1}\sum_{i_2=1}^{N_2} \mu_{A_1^{i_1}}(x_1).\mu_{A_2^{i_2}}(x_2)} \tag{10}$$

where $f(x_1,x_2)$ denotes the output value and $\left\{A_i^1, A_i^2, \ldots, A_i^{N_i}\right\}_{i=1,2}$ are two fuzzy sets with $\left\{\mu_{A_1^{i_1}}(x_1)\right\}_{1\le i_1\le N_1}$ and $\left\{\mu_{A_1^{i_2}}(x_2)\right\}_{1\le i_2\le N_2}$ membership functions defined for the inputs x_1 and x_2. The center of the output fuzzy set, denoted by $\bar{y}^{i_1 i_2}$, is chosen as the output desired value. More information about the derivation steps of the fuzzy system is presented in [17] and [18]. The used fuzzy system can be simply assumed as an approximator. More details about the fuzzy system are out of scope of this paper. See [19] and [20], for a fast introduction to the fuzzy logic. The output of fuzzy system is tuned adaptively according to the buffer size and thereafter buffering delay as

TABLE III
DESIRED CENTRAL VALUES FOR THE OUTPUT OF FUZZY SYSTEM

x_2 \ x_1	**3VL**	**2VL**	**VL**	**L**	**ML**	**M**	**MH**	**H**	**VH**
VH	8	7	6	5	4	3	2	1	0
H	7	6	5	4	3	2	1	0	-1
MH	6	5	4	3	2	1	0	-1	-2
M	5	4	3	2	1	0	-1	-2	-3
ML	4	3	2	1	0	-1	-2	-3	-4
L	3	2	1	0	-1	-2	-3	-4	-5
VL	2	1	0	-1	-2	-3	-4	-5	-6

$$\Delta Q_R = G \cdot f(x_1,x_2) \cdot \frac{R}{S} \tag{11}$$

where G is a constant coefficient typically about 0.3 which defines the gain of feedback control loop.

D. Quality Controller

The quality controller computes an output for each encoder according to the quality of encoded bit streams to balance the quality across the bit streams. To compute the outputs of the quality controller, a reference point based on the temporal average values of PSNR and QP is considered. Then, it is attempted to drive the PSNR of the following pictures to the reference point with a gain proportional to the current deviation from the reference point. Consider the definition of PSNR, for small deviations around the reference point, it can be assumed that the quantization error is proportional to QP, then

$$\frac{\Delta QP}{\Delta \mathrm{PSNR}} \approx \frac{\mathrm{dQP}}{\mathrm{PSNR}} = \frac{\mathrm{QP}}{-C\log e} \tag{12}$$

where C is a constant coefficient. According to this, the output of the quality controller used in (6) is proposed to be computed by

$$\Delta Q_{Qn} = \theta \cdot \bar{Q}\left(\mathrm{PSNR}_n - \overline{\mathrm{PSNR}}\right) \tag{13}$$

where $\bar{Q}$ denotes the average of QP over all pictures in the last encoded super picture. PSNR_n stands for the PSNR of the last encoded picture in the nth bit stream and $\overline{\mathrm{PSNR}}$ is the average of PSNR over all pictures in the super picture. θ is a constant coefficient, typically about 0.03, which is defined experimentally. To suppress the fluctuation in QP results of short-term variations in PSNR, a simple low pass filter (LPF) is used to smooth the variations in $\bar{Q}$ and $\overline{\mathrm{PSNR}}$ before using in (13). The impulse response of the LPF can be expressed as

$$H(z) = \frac{m}{m+1-z^{-1}} \tag{14}$$

where m is a constant value and good results are obtained with $m = 0.5$.

V. SIMULATION RESULTS

The performance of the proposed joint video coding and StatMux method is evaluated through simulations. A number of six known video sequences including *Foreman*, *Carphone*, *New York*, *Football*, *Suzie*, and *Table Tennis* sequences were repeated and concatenated to make long (900 frames, 30 f/s) sequences (S1, S2, S3, S4, S5, and S6; respectively) suitable for the simulations. The provided sequences were encoded in three ways: constant QP, using independent rate controllers, and by the proposed joint rate control algorithm. The target bit rate was set to 300 kb/s for the constant QP case and the independent rate control case and it was set to 1800 kb/s (6 × 300 kb/s) for the joint encoding case. For the independent rate control case, a fuzzy rate control algorithm similar to [10] was used which is optimized for a DVB-H application. To achieve exact average target bit rate in the constant QP case, the video sequence is divided in two sections and the two sections are encoded by two consequent values of QP. The exact target bit rate is achieved by controlling the number of video pictures in each section. The Nokia H.264/AVC codec [12] was used for encoding. Level 3 of the baseline profile, with RDO (Rate Distortion Optimization) was used for the implementation of the three encoding methods. Default values were used for the remaining encoding parameters.

To evaluate the video quality of the proposed joint rate controller, PSNR of the luma component and the standard deviation (StD) of PSNR in each case were measured. Simulation results are presented in Table IV. Note the large variation in the values of PSNR in the constant QP case and the case of independent rate control; while the variation is much smaller in the case of joint encoding. Considering the standard deviation of PSNR as a quality measure, simulation results show that the standard deviation of PSNR decreases from 2.33 dB (on the average over the video sequences) in the independent rate control case to 1.27 dB in the joint encoding case that is close to 0.90 dB in constant QP case. The smaller standard deviation of PSNR means less variation in the quality of encoded video that corresponds to higher visual quality. If the standard deviation of PSNR is considered across the bit streams, it decreases from 3.1 dB in constant QP case to 1.32 dB in the joint encoding case. The lower standard deviation of PSNR across the bit stream means more constant quality across the services. The graphs depicted in Fig. 5 show the PSNR of encoded bit streams in the three cases. The graphs show how well the qualities of joint encoded bit streams are driven close together while they have only smooth variations comparable to the constant QP case.

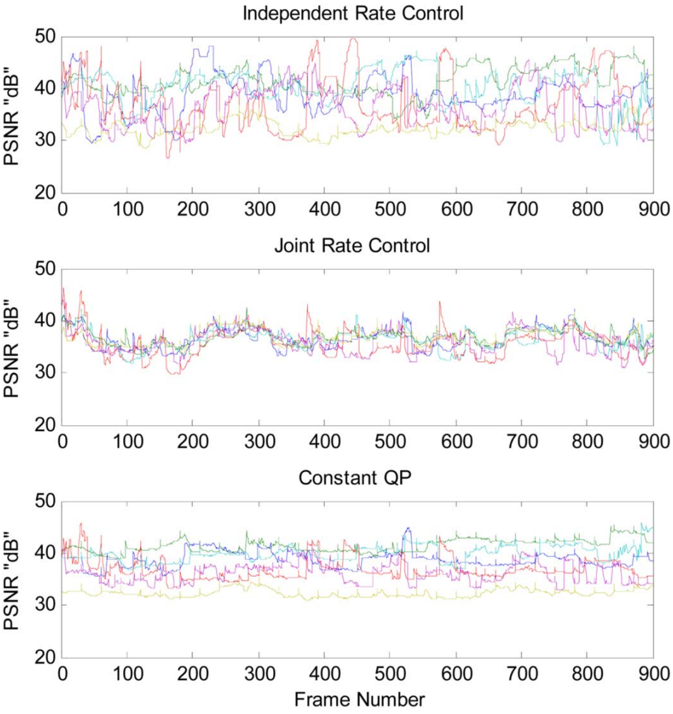

Fig. 5. PSNR graphs of video bit streams encoded by three methods.

TABLE IV
COMPARISON OF PSNR AND THE STANDARD DEVIATION OF PSNR ON 6 VIDEO SEQUENCES ($S1 \ldots S6$) IN THREE ENCODING METHODS INCLUDING INDEPENDENT ENCODING, CONSTANT QP AND JOINT ENCODING.

	Independent Rate Control		**Constant QP**		**Joint Rate Control**	
Seq.	PSNR “dB”	StD PSNR	PSNR “dB”	StD PSNR	PSNR “dB”	StD PSNR
S1	38.16	1.32	38.11	0.73	38.23	1.29
S2	39.93	2.67	39.77	0.70	39.24	1.11
S3	40.04	2.62	39.77	0.71	38.75	0.91
S4	33.07	3.85	32.76	1.92	35.26	1.96
S5	43.09	1.46	43.06	0.58	38.71	1.11
S6	37.93	2.05	37.79	0.77	38.65	1.23
Avr.	38.70	2.33	38.54	0.90	38.14	1.27

To evaluate the proposed joint encoding and StatMux method from the buffering delay point of view, three simulations were run on the encoded bit streams. In first simulation, the encoded bit streams by independent rate controllers are transmitted to the decoder through separate channels defined by DetMux. In the second simulation, the encoded bit streams by independent rate controllers are statistically multiplexed as described in Section III and transmitted to the receivers through a common channel with a bandwidth of 1.8 Mb/s. In the third simulation, the jointly encoded bit streams are transmitted to the receivers through the common channel. In each simulation, the minimum required buffering delay and buffer size in the decoder for zero data drop rate were measured. Simulation results are summarized in Table V. The average (over all sequences) results show that the required buffering delay decreases from 0.23 s in independent encoding and DetMux case to 0.18 s in joint encoding and StatMux case and it decrease to 0.13 s in

TABLE V
BUFFERING DELAY AND BUFFER SIZE OF SIX VIDEO SEQUENCES $(S1 \ldots S6)$ IN THREE CASES INCLUDING INDEPENDENT ENCODING AND DETMUX, INDEPENDENT ENCODING AND STATMUX, AND JOINT ENCODING AND STATMUX

	Independent RC & DetMux		**Independent RC & StatMax**		**Joint Rate Control**	
Seq.	Delay “s”	Buffer “kbit”	Delay “s”	Buffer “kbit”	Delay “s”	Buffer “kbit”
S1	0.18	123	0.14	97	0.17	112
S2	0.24	132	0.13	79	0.14	99
S3	0.20	110	0.13	77	0.14	90
S4	0.14	109	0.12	81	0.32	199
S5	0.26	154	0.14	87	0.11	69
S6	0.33	170	0.13	113	0.20	158
Ave.	***0.23***	***133***	***0.13***	***89***	***0.18***	***121***

the case of independent rate control and StatMux. Moreover, the average required buffer sizes decrease from 133 kbits to 121 and 89 kbits, respectively. Although the independent encoding and StatMux case has provided a lower delay than joint encoding, the overall quality of joint encoding is much better than independent coding. Moreover, if the number of services increases, the joint encoding system provides less delay. Furthermore, using a smaller joint buffer can provide lower delay for the joint encoding case at the expense of more variations in the video quality. The results show how the proposed joint rate control system can decrease the delay and improve the overall quality at the same time.

It is remarkable that the measured decoder buffering delay is only a part the rate-dependent buffering delay in a DVB-H broadcast system. The other parts of the rate-dependent buffering delay also decrease by the proposed method. For example, the Pre-encapsulation Buffering Delay in the IP encapsulator is symmetric with respect to the decoder buffering delay and therefore it will be reduced in a similar fashion.

The computational complexity of the proposed rate controller was compared with the rate control algorithm of the *Joint Model (JM)* [21] reference software of H.264/AVC video coding standard. Note that the proposed rate controller has a granular structure such that it can be used for encoding any number of sources $(n \geq 1)$. In the simulations, the first 100 video frames of four known video sequences, *Foreman*, *Carphone, News*, and *Hal,l* were encoded by two rate controllers with 300 kb/s, 3 f/s, buffer size of 300 kb, and QCIF picture format. The other encoding parameters were used as before. The JM rate controller was tuned to work at the picture level to provide the minimum complexity. The consumed processing times by the rate controllers were measured using the processor clock (Intel Pentium4, 2.8 GHz). To minimize the measuring error results of time sharing operation of the processor, the encoding was repeated five times and the minimum measured value was selected for each sequence. However, the measured values over repetitions are very similar with a small variance. The measured results are shown in Table VI. The average results over the video sequences show that the JM rate controller consumes a processing time of about 384 μs (micro second) in average for each frame while the proposed rate controller consumes a processing time of about 15 μs in average for each frame. According to the provided results, there is a large difference between the complexities of the two controllers. However, in joint video coding, the computational complexities of conventional rate controllers increase proportionally with the number of bit streams but in the proposed method the main part of computational complexity that is related to the fuzzy controller is independent of the number of bit streams. This means that when the number of bit streams increases, the computational complexity of proposed algorithm will decrease even more in comparison with the conventional algorithms.

TABLE VI
COMPARISON OF THE PROPOSED AND JM RATE CONTROL ALGORITHMS IN TERM OF PROCESSING TIME

Sequence	**Processing Time *‘Micro Second’***	
	JM	**Proposed**
Foreman	39327	1537
Carphone	38136	1538
News	38116	1556
Hall	38379	1540
Average Over Sequences	38489	***1543***
Average Per Frame	***384***	***15***

VI. CONCLUSIONS

Different scenarios for encoding and multiplexing of DVB-H services were studied. The advantages of statistical multiplexing in terms of end-to-end delay reduction in a DVB-H broadcast system were evaluated experimentally. The results show that statistical multiplexing has a considerable potential to decrease the end-to-end delay of a DVB-H system. An encoding and statistical multiplexing method was proposed to utilize the advantages of statistical multiplexing in DVB-H. Simulation results show that the proposed method can decrease the end-to-end delay of a broadcast system and improve the overall quality of broadcast services. The proposed technique has a much lower computational complexity compared to existing techniques. Moreover, it has a granular structure such that it can be applied to any number of video services.

REFERENCES

[1] *Digital Video Broadcasting (DVB): Transmission Systems for Handheld Terminals*, ETSI Standard, EN 302 304 V1.1.1, 2004.

[2] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, “DVB-H: digital broadcast services to handheld devices,” *Proc. IEEE*, vol. 94, no. 1, pp. 194–209, Jan. 2006.

[3] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, “Tune-in time reduction in video streaming over DVB-H,” *IEEE Trans. Broadcasting, Special Issue on Mobile Multimedia Broadcasting*, vol. 53, no. 1, pt. 2, pp. 320–328, Mar. 2007.

[4] I. Bouazizi and M. Rezaei, “Approaches for smooth buffering in mobile TV over DVB-H,” in *Proc. 56th Annu. IEEE Broadcast Symp.*, Washington, DC, , Sep. 2006.

[5] *Digital Video Broadcasting (DVB); IP Datacast Over DVB-H: Conent Delivery Protocols*, ETSI Standard, TS 102 472, V1.2.1, 2006.

[6] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: Tradeoffs and potentials," *Proc. IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.

[7] I. Koo, P. Nasiopoulos, and R. Ward, "Joint MPEG-2 coding for multi-program broadcasting of pre-recorded video," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 1999, vol. 4, pp. 2227–2230.

[8] L. Boroczky, A. Y. Ngai, and E. F. Westermann, "Joint rate control with look-ahead for multi-program video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 7, pp. 1159–1163, Oct. 2000.

[9] L. Wang and A. Vincent, "Bit allocation and constraints for joint coding of multiple video programs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 949–959, Sep. 1999.

[10] H. Xiong, J. Sun, S. Yu, C. Luo, and J. Zhou, "Design and implementation of multiplexing rate control in broadband access network TV transmission system," *IEEE Trans. Consumer Electron.*, vol. 50, no. 3, pp. 849–855, Aug. 2004.

[11] *DVB-H Implementation Guidelines*, ETSI Standard, TR 102 377, 2005.

[12] 2005 [Online]. Available: ftp://standards.polycom.com/IMTC_Media_Coding_AG/

[13] M. Rezaei, M. Gabbouj, and I. Bouazizi, "Delay constrained fuzzy rate control for video streaming over DVB-H," in *Proc. IEEE Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, Pasadena, CA, Dec. 2006.

[14] M. Rezaei, S. Wenger, and M. Gabbouj, "Video rate control for streaming and local recording optimized for mobile devices," in *Proc. IEEE Int. Symp. Personal Indoor and Mobile Radio Communications (PIMRC)*, Berlin, Germany, Sep. 2005.

[15] M. Rezaei, M. M. Hannuksela, and M. Gabbouj, "Low-complexity fuzzy video rate controller for streaming," in *Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[16] X. M. Zhang, A. Vetro, Y. Q. Shi, and H. Sun, "Constant quality constrained rate allocation for FGS-coded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 121–130, Feb. 2003.

[17] L. X. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 146–155, May 1993.

[18] L. X. Wang, *Adaptive Fuzzy System and Control: Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[19] L. A. Zadeh, "Soft computing and fuzzy logic," *IEEE Software*, vol. 11, no. 6, pp. 48–56, Nov. 1994.

[20] L. A. Zadeh, *IEEE Comput. Mag.*, vol. 21, 1988.

[21] G. Sullivan, T. Wiegand, and K. P. Lim, "Joint model reference encoding methods and decoding concealment methods," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Document JVT-I049*, San Diego, CA, Sep. 2003.

Mehdi Rezaei (M'04) received the B.S. degree in electronics engineering from Amir Kabir University of Technology (Polytechnic of Tehran), Tehran, Iran, in 1992 and the M.Sc. degree in electronics engineering from Tarbiat Modares University of Tehran in 1996.

He was an Academic Member of the Electrical Engineering Department at the University of Sistan and Balouchestan, Iran, during 1997–2003. He has been a Researcher with the Institute of Signal Processing at Tampere University of Technology, Tampere, Finland, since 2003. His research interests include video encoding and signal processing, variable bit rate video, video rate control, video streaming and communication, mobile TV, and DVB-H. He has published several papers in these fields.

Dr. Rezaei was the recipient of the Nokia Foundation Award in 2005 and in 2006.

Imed Bouazizi (M'03) received the Diplom. Inform. (equivalent to M.S.) degree in computer sciences from Technical University of Braunschweig, Braunschweig, Germany, and the Ph.D. degree from the University of Aachen, Aachen, Germany.

He is a Senior Researcher at the Nokia Research Center. His research field covers all aspects of multimedia communication and video compression, with a strong focus on multicast and broadcast technologies such as DVB-H. He has contributed to several standardization efforts in the most relevant organizations such as DVB, 3 GPP, and the IETF. He also holds several patents and patent applications in related areas and applications.

Moncef Gabbouj (SM'95) received the B.S. degree in electrical engineering in 1985 from Oklahoma State University, Stillwater, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1986 and 1989, respectively.

He is currently a Professor and Head of the Institute of Signal Processing at Tampere University of Technology, Tampere, Finland. He is the co-founder and past CEO of SuviSoft Oy Ltd. From 1995 to 1998, he was a Professor with the Department of Information Technology, Pori School of Technology and Economics, and during 1997 and 1998 he was a Senior Research Scientist with the Academy of Finland. From 1994 to 1995, he was an Associate Professor with the Signal Processing Laboratory of Tampere University of Technology, Tampere, Finland. From 1990 to 1993, he was a Senior Research Scientist with the Research Institute for Information Technology, Tampere, Finland. He is co-author of over 280 publications. He has been involved in several past and current Eurpean Union Research and education projects and programs, including ESPRIT, HCM, IST, COST, Tempus, and Erasmus. He also served as Evaluator of IST proposals, and Auditor of a number of ACTS and IST projects on multimedia security, augmented and virtual reality, image and video signal processing. He is the Director of the International University Programs in Information Technology and vice member of the Council of the Department of Information Technology at Tampere University of Technology. He is also the Vice-Director of the Academy of Finland Center of Excellence SPAG, Secretary of the International Advisory Board of Tampere International Center of Signal Processing, TICSP and member of the Board of the Digital Media Institute. He serves as Tutoring Professor for Nokia Mobile Phones Leading Science Program (1998-2001). His research interests include multimedia content-based analysis, indexing and retrieval; nonlinear signal and image processing and analysis; and video processing and coding.

Dr. Gabbouj is a Honorary Guest Professor of Jilin University, China (2005–2010), a Distinguished Lecturer for the IEEE Circuits and Systems Society, and Past-Chairman of the IEEE-EURASIP Nonlinear Signal and Image Processing Board. He was Chairman of the Algorithm Group of the EC COST 211 quat. He served as Associate Editor of the IEEE Transactions on Image Processing and was Guest Editor of the European journal *Applied Signal Processing* (Image Analysis for Interactive Multimedia Services, Part I in April and Part II in June 2002) and *Signal Processing*, Special Issue on Nonlinear Digital Signal Processing (August 1994). He is the past Chairman of the IEEE Finland Section and past Chair of the IEEE Circuits and Systems Society, Technical Committee on Digital Signal Processing, and the IEEE SP/CAS Finland Chapter. He was also Chairman of CBMI 2005, WIAMIS 2001, and the TPC Chair of ISCCSP 2004, CBMI 2003, EUSIPCO 2000, NORSIG 1996, and the DSP Track Chair of the 1996 IEEE ISCAS. He is a member of EURASIP AdCom and also served as Publication Chair and Publicity Chair of IEEE ICIP 2005 and IEEE ICASSP 2006, respectively. He was the recipient of the 2005 Nokia Foundation Recognition Award and co-recipient of the Myril B. Reed Best Paper Award from the 32nd Midwest Symposium on Circuits and Systems and co-recipient of the NORSIG 94 Best Paper Award from the 1994 Nordic Signal Processing Symposium. He is a member of Eta Kappa Nu, Phi Kappa Phi, and the IEEE Signal Processing and Circuits and Systems societies.