Rate-Distortion-Complexity Modeling for Network and Receiver Aware Adaptation

Mihaela van der Schaar, *Senior Member, IEEE,* and Yiannis Andreopoulos, *Student Member, IEEE*

***Abstract*—Existing research on Universal Multimedia Access has mainly focused on adapting multimedia to the network characteristics while overlooking the receiver capabilities. Alternatively, part 7 of the MPEG-21 standard entitled Digital Item Adaptation (DIA) defines description tools to guide the multimedia adaptation process based on both the network conditions and the available receiver resources. In this paper, we propose a new and generic rate-distortion-complexity model that can generate such DIA descriptions for image and video decoding algorithms running on various hardware architectures. The novelty of our approach is in *virtualizing complexity*, i.e., we explicitly model the complexity involved in decoding a bitstream by a generic receiver. This generic complexity is translated dynamically into "real" complexity, which is architecture-specific. The receivers can then negotiate with the media server/proxy the transmission of a bitstream having a desired complexity level based on their resource constraints. Hence, unlike in previous streaming systems, multimedia transmission can be optimized in an integrated rate-distortion-complexity setting by minimizing the incurred distortion under joint rate-complexity constraints.**

***Index Terms*—Complexity modeling, MPEG-21 digital item adaptation, rate-distortion optimization, video streaming.**

I. Introduction

TRADITIONALLY, multimedia compression and streaming have been studied within the rate-distortion theory framework that defines tradeoffs between information rate and distortion. Nonscalable bitstream switching, adaptive rate scaling, transcoding, scalable coding, distortion-optimized packet scheduling, network-adaptive source/channel coding, multiple description coding, etc. [1], have been developed to address real-time adaptation of multimedia content at the server or on-the-fly at a proxy based on the network conditions. However, in most cases, these network-centric approaches neglect the user experience, as well as the capabilities and resource constraints of the receiver (e.g., display size, processing power, battery-life etc.).

Part 7 of the MPEG-21 standard, entitled Digital Item Adaptation (DIA), has defined a set of description tools for adapting multimedia based on the user characteristics, terminal capabilities, network characteristics and natural environment characteristics [2], [3]. This paper introduces a new rate-distortion-complexity (R-D-C) model that can assist the MPEG-21 DIA engine to *explicitly* consider the resources available at the receiver. These include hardware resources such as memory, processors, functional units, instruction, and data memory bandwidths and limits on the power dissipation.

The focus of this paper is not on system-specific complexity or power optimization since these topics have already been thoroughly studied for different multimedia codecs (e.g., see the work for software implementations of the AVC codec [4], [5], the work for Motion JPEG2000 [6], and the work for MPEG-21-based scalable wavelet decoding [7]). Instead, the novelty of our approach is twofold. Firstly, in this paper we introduce the concept of "virtual" decoding complexity and determine a general R-D-C model that can be easily applied to a variety of existing and upcoming image and video compression schemes. Secondly, unlike the MPEG-4 Video Complexity Verifier (VCV) [16], [17] that determines whether the decoding resources fit within a certain profile (which corresponds to maximum allowable decoder resources), we consider *average* decoding complexities estimated using a model-based approach that considers the decoding algorithm implementation, as well as the transmission bit-rate and content characteristics. While worst-case bounds on complexity are extremely important for dedicated hardware implementations (e.g., application-specific integrated circuits), they are not very meaningful for next generation programmable architectures that can support multi-fidelity algorithms by allowing dynamic resource allocation. Examples of such architectures include energy-adjustable processors with dynamic frequency and voltage scaling and reconfigurable architectures.

In the proposed framework, the server generates a platform-independent model quantifying a set of *generic complexity metrics* (GCMs) for decoding/streaming. We propose a general *R-D-C modeling methodology* to generate the GCMs for different operational R-D points.

The (pre-computed) GCMs are then mapped into *real complexity metrics* (RCMs) that explicitly consider the specific terminal architectures and available resources. Consequently, multimedia bitstreams are correspondingly adapted at the server, proxy or receiver based on the determined RCMs.

This paper is organized as follows. In Section II, we review previous work in the area of complexity-scalable and complexity-adaptable systems and explain the advances offered by this paper. Section III presents a general methodology for constructing R-D-C models. We also introduce the GCM

Manuscript received May 10, 2004; revised August 27, 2004. This work was presented in part at the Global Telecommunications Conference, Dallas, TX, December 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Fernando M. B. Pereira.

M. van der Schaar is with the Department of Electrical and Computer Engineering, University of California Davis, Davis, CA 95616-5294 USA (e-mail: mvanderschaar@ece.ucdavis.edu).

Y. Andreopoulos is with the Department of Electrical and Computer Engineering, University of California Davis, Davis, CA 95616-5294 USA, on leave from the Department of Electronics and Information Sciences, Vrije Universiteit Brussel (Free University of Brussels), B-1050 Brussels, Belgium (e-mail: yandreop@etro.vub.ac.be).

Digital Object Identifier 10.1109/TMM.2005.846790

concept, which enables the definition of common (generic) complexity metrics across different classes of receivers. Section IV illustrates how the proposed R-D-C models can be used for multimedia adaptation based on network characteristics and terminal capabilities. In Section V, indicative simulation results and experiments using the proposed R-D-C models are presented. Finally, Section VI concludes the paper and discusses our future research.

II. Related Prior Work

Previous related research efforts can be classified into three categories:

- complexity-scalable multimedia encoding and decoding algorithms;
- receiver-driven multimedia streaming based on channel and end-device characteristics;
- generic and real complexity metrics and models for multimedia coding and streaming introduced by van der Schaar *et al.* [12], [14].

A number of authors have considered complexity-scalable coders by focusing on various aspects. For example, van der Schaar and de With [8] considered scalable memory complexity reductions by recompressing I- and P- frames at the decoder prior to motion-compensated prediction. An audio decoder with computational scalability has been introduced by Argenti *et al.* [9] by considering a partial reconstruction of the signal spectrum. Finally, decoding of images with scalable complexity has been systematically studied by Pan and Ortega [10].

In the area of adaptive multimedia streaming, McCanne *et al.* [11] proposed a receiver-driven multicast system that allows receivers to selectively subscribe to multiple channels based on their available bandwidth. Van der Schaar [15] introduced a generic system for video compression and transmission based on channel and receiver characteristics in order to accommodate a set of rate, distortion and capability constraints imposed by the receiver. Chen *et al.* [13] discussed a practical implementation of such a system by utilizing a multi-track hinting system, originally introduced by Li and van der Schaar [31], for multimedia adaptation based on complexity. In a similar context, van der Schaar *et al.* [14] presented a simple R-D-C adaptation mechanism for wavelet-based video decoding based on the number of decoded nonzero coefficients used prior to the inverse discrete wavelet transform. Finally, a real time video streaming system based on the complexity modeling framework of this paper was presented by van der Schaar *et al.* [12].

In this paper, our focus will not be on complexity-scalable modifications of existing video decoders. In contrast, without any attempt to modify or optimize the decoder implementation, we build upon the original proposal for R-D-C video adaptation [15] and attempt to *model* complexity based on the compressed source characteristics and the decoding platform capabilities. In addition, we deploy the proposed complexity model to drive complexity adaptation in a realistic multimedia testbed.

III. Rate-Distortion-Complexity Models

Recently, rate-distortion theory was extended to complexity-distortion theory and the complexity scalability of several simple algorithms (e.g., searching algorithms) has been investigated [18]. To enable on-the-fly R-D-C adaptation within the MPEG-21 DIA framework, a practical R-D-C model is required that relates the various operational R-D points (corresponding to different substreams) to their corresponding decoding complexity. Our focus will be on modeling the generic complexity of multimedia decoding algorithms that does not consider the specific receiver features, capabilities and instantaneous resources.

A. Generic Modeling of Complexity

In order to represent at the server side different receiver architectures in a generic manner, we will deploy a concept that has been successful in the area of computer systems, namely, a virtual machine. We assume an abstract receiver referred to as a generic reference machine (GRM). This is representative of the computation and resource models of the receiver architectures in use. The GRM can be viewed as a basic pipelined RISC machine [19]. Assuming the GRM as the target receiver, we will develop an abstract complexity measure to quantify the decoding complexity of multimedia bitstreams.

The key idea of the proposed paradigm is that the same bitstream will require/involve different resources/complexities on various receivers. Given the number of factors that influence the complexity of the receiver, it is impractical to determine at the server side the *specific* (real) complexity for every possible receiver architecture. Consequently, we adopt a *generic complexity model* that captures the abstract/GCMs of the employed decoding or streaming algorithm depending on the content characteristics and transmission bit-rate. GCMs are derived by computing the average number of times the different GRM-operations are executed. In this paper, we consider a simple GRM that supports the following operations[1]:

$$\mathrm{op} = \{\mathrm{add}, \mathrm{multiply}, \mathrm{assign}\}. \tag{1}$$

In DIA, the AdaptationQoS tool [2] defines adaptation units (AUs) as a group of video macroblocks, an entire video frame, a certain resolution of a frame, a group of pictures (GOP) etc. The GCMs necessary for the decompression and streaming can be transmitted to the receiver at different granularities (e.g., for each functional unit, subunit etc.) and for varying-size AUs [2]. In general, finer granularity allows better control of the adaptation, but this may come at the expense of an increased communication and computational overhead.

Several illustrative examples on how R-D adaptation can be performed in the AdaptationQoS framework can be found in related literature [20]. In this paper, we develop models to be used in this framework for complexity adaptation. Although our experiments will be based on a scalable video coder, it is important to notice that the proposed models apply more broadly, e.g., to conventional nonscalable video coding schemes [21], [22], as well as multiple description coding schemes. The only difference is that, to create a set of alternative bitstreams resulting in

[1]More sophisticated GRMs can be defined to facilitate better mapping of GCMs to architecture-dependent resources, e.g., different data and memory types, word lengths, etc. However, this involves more complex R-D-C modeling, GCM to RCM mapping, and bitstream adaptation mechanisms.

different rate, distortion and complexity tradeoffs, multiple encodings/transcodings of the same content are typically required, which incurs higher computational load for the server.

We assume that each video GOP is partitioned into N independently-coded adaptation units. For example, to provide efficient resolution scalability, the maximum size of an AU is usually bounded to be an entire resolution level of a given intra- or inter-frame in the GOP. Let the set of AUs that correspond to the decoded resolution and frame rate of a GOP be denoted as $\{b_1, b_2, \ldots, b_N\}$. Each independent AU $b_i, 1 \leq i \leq N$, is associated with a set of rate-distortion points $\{R_i^{j(i)}, D_i^{j(i)}\}$ with $j(i)$ indicating the corresponding bitstream-adaptation point. Within the AdaptationQoS tool of MPEG-21 DIA, the rate-distortion points are termed as *dependent IOPins* [20] since they depend on the number of permissible decoding parameters. This dependency stems from the fact that feasible adaptation points, which are referred to as *free IOPins* [20], can be derived at different spatio-temporal resolutions.

An optimization that aims to minimize the overall distortion in the GOP under a rate-constraint $R_{\max}$ can be stated as

$$\{j_r^*(i), \lambda_r^*\}_{\forall b_i} = \arg\min_{j(i),\lambda} \left\{ \sum_{i=1}^{N} \left(D_i^{j(i)} + \lambda \cdot R_i^{j(i)} \right) \right\} : R_{\text{GOP}} \leq R_{\max}. \quad (2)$$

The Lagrangian multiplier λ must be adjusted until the value $\lambda = \lambda_r^*$ is found where the rate corresponding to the selected points $j_r^*(i)$ is (approximately) equal to $R_{\max}$.

B. Proposed Generic Complexity Model

To each possible bistream adaptation point formulated by the set of solutions of (2), an associated complexity metric can be defined. We illustrate here how a generic complexity model can be built for video decoders employing motion compensation and transform coding. Similar models can also be build for alternative coding and/or streaming algorithms.

The proposed framework is inspired by the work of He and Mitra [21] on rate-control for image and video compression. Our approach models the expected decoder complexity with an AU-level granularity based on source (video-data) characteristics as well as implementation-related features. For a transform-based motion-compensated video coding scheme, the complexity of decoding is primarily determined by:

- the intra- and inter-frame decoding and inverse transform;
- the motion-compensation process[2].

For each AU b_i, we define the following *complexity-function variables*, corresponding to these decoder operations:

- the percentage of nonzero (decoded) transform coefficients, denoted by $p_T(i)$;
- the percentage (per pixel) of decoded motion vectors out of the maximum number of possible motion vectors (hypotheses) provided by the utilized motion model, denoted by $p_M(i)$.

[2]Although the decoding of the motion-vector information also contributes in the complexity profile of the decoding process, with the exception of extremely low bitrates, the effect of this process in the complexity of practical video decoders can be considered to be negligible.

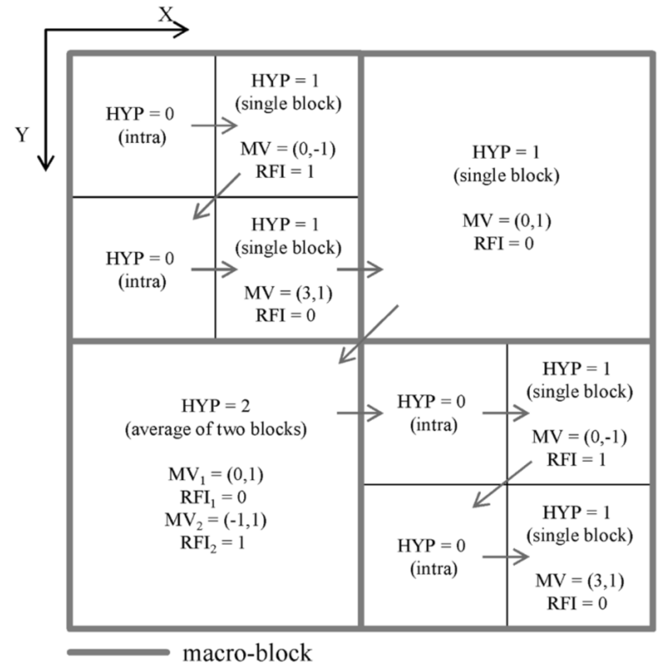


Fig. 1. Motion estimation with variable block-size multihypothesis prediction [23]. Notations: HYP is the number of motion-vectors (hypotheses) used for the prediction of each subblock ($\text{HYP} = 0$ denotes intra-prediction); $\text{MV}_h, 1 \leq h \leq \text{HYP}$, is the h-th motion vector; RFI_h represents an index to the fractional-pixel interpolated position for the h-th motion vector.

The purpose of these variables is not to encapsulate the input source characteristics but rather to represent the *underlying mechanism* based on which the input source characteristics can vary. It was already shown that a model that depends on these variables can be generic and yet efficiently predict the variations of the compressed source data, e.g., see the model of He and Mitra [21], as well as the use of nonzero transform coefficients in the VCV buffer model [17], [16].

Our motivation behind the choice of p_T comes from the fact that, in the decoding process, particular processing operations are activated only when a significant (nonzero) coefficient is found. For example, during the decoding process of wavelet-based embedded codecs, for each processed bitplane, the transform areas with insignificant (zero) coefficients are either skipped or processed in a uniform way by performing a fixed operation such as decoding a nonsignificance symbol. On the other hand, more complex operations such as sign decoding, list management and coefficient refinement operations occur at the areas where coefficients are found to be significant. We refer to the JPEG-2000 algorithm [24] as a practical test case. Similar observations hold for the decoding of other wavelet-based and DCT based image-coding schemes.

Concerning motion information, in state-of-the-art video coding systems, motion is adaptively estimated using a system with variable block-size multihypothesis prediction [22]. An example of such a system is pictorially demonstrated in Fig. 1, [23]. Depending on the motion characteristics, the motion estimation algorithm partitions each macroblock adaptively and a variable number of motion vectors are assigned to the pixels belonging to each macroblock area. However, due to complexity and bandwidth limitations, there always exists a maximum number of motion vectors that can be associated with each pixel in a given frame. Consequently, at the pixel-level, the motion vectors associated with each error frame represent a percentage p_M of the maximum number of the vectors that represent the maximum-density motion field. As a result, the number of arithmetic operations and memory accesses during motion compensation depends on p_M.

For each AU b_i, the proposed complexity functions can now be defined as $W^{\mathrm{op}}(p_T(i), p_M(i))$, with the superscript op given in (1). This estimation of $W^{\mathrm{op}}(p_T(i), p_M(i))$ is based on a decomposition in a set of basis functions that characterize the complexity of the different parts of the motion-compensated wavelet decoding system. These basis functions can be considered as high-level descriptors of the complexity characteristics of each part of the entire decoding system. In particular, we introduce the *texture-related complexity basis-function*, $\mathcal{T}(p_T(i))$, and three *motion-related complexity basis-functions*, $M_F(p_M(i)), M_S(p_M(i))$, and $M_Z(p_M(i))$, which relate to the operation modes associated with the motion-compensation process. Their precise definitions will be given in Section III-C.

For a given AU b_i, the complexity functions $W^{\mathrm{op}}(p_T(i), p_M(i))$ are formulated as

$$\begin{aligned} &W^{\mathrm{op}}(p_T(i), p_M(i)) \\ &\quad = A^{\mathrm{op}}(p_T(i)) \cdot \mathcal{T}(p_T(i)) + B^{\mathrm{op}}(p_M(i)) \cdot M_F(p_M(i)) \\ &\quad + C^{\mathrm{op}}(p_M(i)) \cdot M_S(p_M(i)) \\ &\quad + D^{\mathrm{op}}(p_M(i)) \cdot M_Z(p_M(i)) \end{aligned} \tag{3}$$

where $A^{\mathrm{op}}(p_T(i))$ is the texture-related complexity-decomposition coefficient, and $B^{\mathrm{op}}(p_M(i)), C^{\mathrm{op}}(p_M(i)), D^{\mathrm{op}}(p_M(i))$ are the motion-related complexity-decomposition coefficients. For each AU of the input video, the complexity basis functions provide a high-level estimation of the data-dependent computational resources that are required for the processing of its associated bitstream. They are varying solely based on the source characteristics. On the other hand, the texture- and motion-related complexity-decomposition coefficients are dependent on the decoding algorithm and its implementation architecture. Once the algorithm (and its implementation) is fixed for a given transmission scenario they can be determined offline by using a number of training sequences and profiling results from the specific decoder implementation architecture. Finally, if intra-frame coding algorithms are considered (e.g., Motion JPEG), the motion-related complexity decomposition coefficients and basis functions are annulated and hence, $\forall i : p_M(i) \equiv 0$.

Based on (3), the complexity-decomposition functions for a GOP consisting of N AUs is given by

$$W_{\mathrm{GOP}}^{\mathrm{op}}(p_T, p_M) = \sum_{i=1}^{N} W^{\mathrm{op}}(p_T(i), p_M(i)) + E^{\mathrm{op}}(\bar{p}_T, \bar{p}_M) \tag{4}$$

where $E^{\mathrm{op}}(\bar{p}_T, \bar{p}_M)$ can be interpreted as the basic decoding complexity per GOP, which varies based on its average texture and motion content, represented by

$$\bar{p}_T = \frac{1}{N}\sum_{i=1}^{N} p_T(i), \quad \bar{p}_M = \frac{1}{N}\sum_{i=1}^{N} p_M(i).$$

In typical scenarios, this factor represents the cumulative effect in complexity of decoding operations that are almost data-independent, such as the inverse transform (IDCT or IDWT). Conceptually, (4) is analogous to a frequency-decomposition of a signal. In this case, $E^{\mathrm{op}}(\bar{p}_T, \bar{p}_M)$ can be seen as the "DC" component of the decomposition.

C. Basis-Functions for the Complexity Decomposition

The texture-related complexity basis-function $\mathcal{T}(p_T(i))$ of an AU b_i is the function associated with the number of operations needed to decode a transform representation with a percentage of $p_T(i)$ nonzero coefficients. We define $\mathcal{T}(p_T(i))$ as [21]

$$\mathcal{T}(p_T(i)) = \frac{1}{X_l \cdot Y_l} \sum_{k=0, c_k(p_T(i)) \neq 0}^{X_l \cdot Y_l} \left(\lfloor \log_2 |c_k(p_T(i))| \rfloor + 2 \right) \tag{5}$$

where $c_k(p_T(i))$ represents the decoded value of the k-th transform coefficient of an AU that reconstructs $X_l \times Y_l$ pixels of the input video frame. When decoding stops at resolution l (with $l = 0$ being the original resolution), we have

$$X_l \cdot Y_l = 4^{-l} X_{\mathrm{or}} \cdot Y_{\mathrm{or}}$$

where the original frame has $X_{\mathrm{or}} \times Y_{\mathrm{or}}$ pixels. Equation (5) shows that the decoded transform-coefficient values c_k are a function of $p_T(i)$. In embedded coding, the coefficient values c_k of each AU b_i are a function of the number of decoded (integer or fractional) bitplanes $q(i)$. In typical transform-coding schemes, $p_T(i)$ is monotonically-increasing with $q(i)$. As a result, there is a one-to-one mapping between $p_T(i)$ and $q(i)$. Hence, c_k is also a function of $p_T(i)$. We note here that in the generic case of nonembedded coding, e.g., quantization of DCT coefficients, (5) can be applied as well [21].

Equation (5) determines the sum of the magnitudes and sign information of the nonzero transform coefficients. This is representative for the complexity variations associated with classical transform-based coding schemes.

As explained before, the complexity of motion compensation for each AU b_i is expected to be proportional to the percentage $p_M(i)$ of the existing motion vectors per pixel. This is modeled by the $M_F(p_M(i))$, which is defined as

$$M_F(p_M(i)) = v \cdot p_M(i) \cdot 4^{-l} \tag{6}$$

where v represents the maximum number of motion vectors (hypotheses) associated with each of the $X_l \times Y_l$ pixels of the decoded AU b_i (see Fig. 1).

The complexity of motion compensation is also related to the number of motion vectors associated with fractional-pixel positions. Fractional-pixel accurate motion compensation typically involves interpolation operations with a filter kernel approximating the sinc function, hence requiring additional processing operations. This is modeled by the $M_S(p_M(i))$ function, defined as

$$M_S(p_M(i)) = M_F(p_M(i)) \cdot s(p_M(i)) \tag{7}$$

where $s(p_M(i))$ is the percentage of existing vectors associated with a fractional-pixel position. For each AU b_i, we define this percentage as a function of $p_M(i)$ since, in our experimentation over a large set of sequences encoded using different settings, it was found that, for block-based motion estimation, this percentage is monotonically increasing with $p_M(i)$. Finally, the additional complexity of advanced techniques using overlapped block motion compensation or deblocking is modeled by the $M_Z(p_M(i))$ function. The application of such techniques strongly relates to the local differences between neighboring motion vectors. Specifically, the definition of $M_Z(p_M(i))$ is based on the following steps:

The motion vectors are organized in a 1-D array. All the macroblocks are scanned in a raster order, while motion vectors inside the macroblock are parsed in a zig-zag scan order, as shown with the arrows of Fig. 1.

- For every motion vector $v_m = (x_m, y_m)$ that points to reference frame r_m, where m is the index in the 1-D array, $(1 \leq m \leq M_F(p_M(i)) \cdot X_l \cdot Y_l)$ calculate

$$d_m = (|x_m - x_{m-1}|, |y_m - y_{m-1}|)$$

where we preset $v_0 = (0, 0)$.
- Let $t_m = \{1\text{: if } \|d_m\| \geq Q \text{ or } r_m \neq r_{m-1}; 0\text{: otherwise}\}$, where Q represent a pre-determined threshold.

We define $M_Z(p_M(i))$ as:

$$M_Z(p_M(i)) = \sum_{m=1}^{M_F(p_M(i)) \cdot X_l \cdot Y_l} t_m. \quad (8)$$

The basis functions of (5)–(8) can be calculated dynamically during encoding for each AU b_i. In order to avoid any increase in the encoding computational complexity, statistical methods could be adopted to calculate these basis functions based on their observed properties, following techniques similar to the ones used by He and Mitra [21]. In practice, the increase in encoder complexity was found to be negligible in comparison to the motion estimation complexity and the direct calculation of the basis functions provides a higher modeling accuracy. Having pre-determined values for the complexity-decomposition coefficients, the generated basis-function values for each adaptation point are used in (3) to estimate the GCMs.

D. Estimation of the Complexity Decomposition Coefficients

In this section, we outline the practical method we used to estimate the values of the complexity-decomposition coefficients of (3) for each op of (1). This estimation is based on a training algorithm. For this purpose, we chose 7 CIF video sequences (“Coastguard”, “Foreman”, “Stefan”, “Tempete”, “Paris”, “Football”, “News”—the first 48 frames of each, which correspond to the equivalent of one GOP in the utilized codec [25]) that were representative of a large variety of content. For each AU $b_i, 1 \leq i \leq N$, of each sequence, the values of the basis functions $T(p_T(i)), M_F(p_M(i)), M_S(p_M(i)), M_Z(p_M(i))$ were calculated for a representative set of values of $p_T(i), p_M(i)$. This range of values was selected so that the corresponding bitrate for the texture and motion-vector information fits in the practical bitrate regimes of CIF video sequences. Naturally, the use of more sequences for the training stage and a high granularity in the regime of practical values of $p_T(i), p_M(i)$ helps in increasing the accuracy of the complexity-modeling process. In practice, the estimation of the values of the complexity-decomposition coefficients was performed by the following process:

The encoding algorithm was executed multiple times for each sequence, each time enforcing the number of motion vectors that corresponded to each AU b_i to be equal to a pre-determined value that corresponds to the specific value p_M^e that is of interest, i.e., $\forall i$: $p_M(i) = p_M^e$.

The bitstream extraction and decoding occurred by selectively decoding a number of bitplanes for the texture information of each AU b_i that corresponds to the specific value p_T^e that is of interest, i.e., $\forall i$: $p_T(i) = p_T^e$.

In this way, for each pair $\{p_T^e, p_M^e\}$ from the set of values that spans the bitrates of interest, based on (3) and (4), we have

$$W_{\text{GOP}}^{\text{op}}(p_T^e, p_M^e) = N \cdot W^{\text{op}}(p_T^e, p_M^e) + E^{\text{op}}(p_T^e, p_M^e). \quad (9)$$

The actual values of $W_{\text{GOP}}^{\text{op}}(p_T^e, p_M^e)$ were (off-line) measured by profiling the software implementation of the video decoder for the set of operations defined by (1). Hence, the experimentally-derived GCMs $W_{\text{GOP}}^{\text{op}}(p_T^e, p_M^e)$ depend on the software implementation of the decoder. Based on the profiling results and on the complexity basis functions (which were calculated using (5)–(8) with $p_T(i) = p_T^e$ and $p_M(i) = p_M^e$ for every i), the values of $A^{\text{op}}(p_T^e), B^{\text{op}}(p_M^e), C^{\text{op}}(p_M^e), D^{\text{op}}(p_M^e)$, and $E^{\text{op}}(p_T^e, p_M^e)$ of (9) were determined using multiple linear regression. Once computed for each $\{p_T^e, p_M^e\}$, these values were kept in lookup tables for the actual GCM estimation during the streaming process.

To summarize, the previously-described off-line training stage only involves the adjustment of the number of motion vectors and the number of significant coefficients of each frame to predetermined values that correspond to each pair $\{p_T^e, p_M^e\}$. The first task is typically accomplished by iteratively adjusting the Lagrangian-based motion-vector pruning scheme of the encoder [22], [25]. The latter is easily performed at the bitstream extraction stage.

We note here that the profiling of the decoder software implementation is generic and can typically be performed using automated analysis tools (e.g., recent work [26] for the memory-related operations). This is an important aspect since, in this way, if a different decoder software-implementation is to be used, the off-line training stage can be re-performed with the new software and different sets of complexity decomposition coefficients can be derived and used in the model.

Finally, the usage of the proposed model during the actual bitstream adaptation process in the streaming server is straightforward. For each AU b_i, depending on the $\{p_T(i), p_M(i)\}$, the complexity-decomposition functions are calculated [see (5)–(8)]. Then, the corresponding complexity decomposition coefficients are found by using lookup tables and rounding the values $\{p_T(i), p_M(i)\}$ to the closest lookup-table entry. At this stage, using (3) and (4) for the current GOP, the model-based estimation of the GCMs takes place.

IV. A Practical Scenario for Complexity-Driven Adaptation

In this section, we show how the proposed R-D-C model allows for dynamic adaptation to network characteristics and terminal capabilities. As mentioned in the introduction, the mapping of GCMs into RCMs can be performed either at the server or the receiver. Fig. 2 pictorially represents an example of the proposed R-D-C bitstream adaptation.

The GCMs $\{W_{\rm op}\}_i^{j(i)}$ are determined at encoding time for each AU, based on the values of $p_M(i)$ and $p_T(i)$. The GCMs are then transmitted to a *complexity mapper*[3], which is aimed at translating the GCMs into RCMs by taking into account a number of factors specific to the receiver architecture and processing platform. These factors include the instruction set of the underlying processor, the data-types supported within the processor, the number of functional units (including hardwired co-processors), the memory hierarchy and input-output processors, the available energy resources and special resources such as FIFOs or other buffers [6]. To determine the available resources, a *resource monitor* can be implemented at the receiver that maintains the current utilization of various resources especially memory, energy etc. The complexity mapper and resource profiler can be implemented in a variety of ways. The reader is referred to recent work [27]–[29] for some illustrative examples for this topic. Based on the generated RCMs that entail the number of different operations that can be supported at a specific time, the receiver metadata is generated. Within the DIA standard, the generation of such metadata will utilize user environment description (UED) tools such as the CPUBenchmark, PowerCharacteristics, DisplayCapabilities, etc. With such descriptions, complexity adaptation at the server/proxy is enabled. Moreover, other metadata (e.g., MPEG-7 content descriptors) can also be generated to assist the Decision Taking Engine that derives the adapted multimedia bitstream that will be streamed to the receiver.

As a result, for each AU b_i and at each possible adaptation point $j(i)$, the receiver formulates the RCM-based metric as

$$C_i^{j(i)} = \sum_{\forall \rm op} \mathcal{L}\left(\{W^{\rm op}\}_i^{j(i)}\right) \tag{10}$$

where $\mathcal{L}(\cdot)$ represents the mapping operation from GCMs to RCMs, which, in the framework of MPEG-21 DIA, involves descriptors from the AdaptationQoS tool. Notice that the RCMs of (10) are additive for all the AUs of each GOP, i.e., $C_{\rm GOP} = \sum_{i=1}^N C_i^{j(i)}$. Let $C_{\max}$ denote the upper bound for the complexity of the receiver in terms of RCMs per GOP. The adaptation points $j(i)$ for each GOP can be formulated as the parameters to a complexity-constrained optimization problem

$$\begin{aligned}&\{j_c^*(i), \lambda_c^*\}_{\forall b_i}\\ &= \arg\min_{j(i),\lambda}\left\{\sum_{i=1}^N \left(D_i^{j(i)} + \lambda\cdot C_i^{j(i)}\right)\right\} : C_{\rm GOP} \le C_{\max}\end{aligned} \tag{11}$$

where the Lagrangian multiplier λ must be adjusted until the value $\lambda = \lambda_c^*$ is found where the RCM-based complexity corresponding to the selected bitstream adaptation points $j_c^*(i)$ is equal to the RCM constraint $C_{\max}$. The optimization of (11) determines the adaptation points for each AU incurring the minimum distortion under the pre-defined constraint $C_{\max}$.

[3]As mentioned in the introduction, the complexity mapper can be located at the receiver (as depicted in Fig. 2), or at the server/proxy.

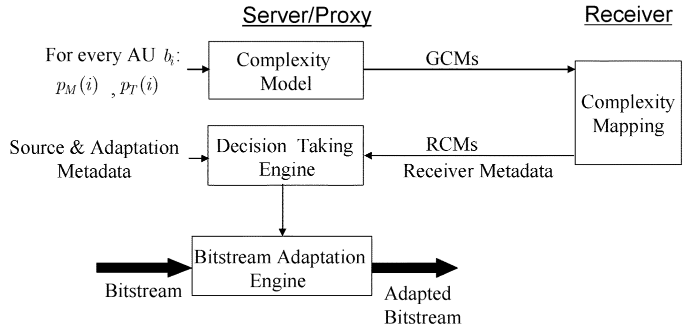


Fig. 2. Example of R-D-C driven adaptation.

In general, the solutions of (2) and (11) (rate- and complexity-constrained optimizations, respectively) will produce *different* truncations for each resolution and frame-rate. However, for practical applications of video streaming, both the rate and complexity constraints must be satisfied, i.e., the solution $\{j^*(i), \lambda^*\}_{\forall b_i}$ must satisfy $C_{\rm GOP} \le C_{\max}$ *and* $R_{\rm GOP} \le R_{\max}$. The optimization problem is then expressed as

$$\begin{aligned}&\{j^*(i), \lambda_r^*, \lambda_c^*\}_{\forall b_i}\\ &\quad = \arg\min_{j(i),\lambda_r,\lambda_c}\left\{\sum_{i=1}^N\left(D_i^{j(i)} + \lambda_r\cdot R_i^{j(i)} + \lambda_c\cdot C_i^{j(i)}\right)\right\}:\\ &\quad R_{\rm GOP}\le R_{\max} \quad \text{and} \quad C_{\rm GOP} \le C_{\max}.\end{aligned} \tag{12}$$

Since the solution of (12) involves searching among various values of λ_r, λ_c, the optimization problem can be efficiently solved by first excluding all the truncation points $j_n(i)$ that do not produce monotonically-decreasing slope values (for each b_i). Effectively, a point $j_n(i)$ is excluded if $\lambda_r^{n+1} > \lambda_r^n$ or $\lambda_c^{n+1} > \lambda_c^n$, with $\lambda_r^{n+1}, \lambda_c^{n+1}$ defined as

$$\lambda_r^{n+1} = \frac{D_i^{j_n(i)} - D_i^{j_{n+1}(i)}}{R_i^{j_{n+1}(i)} - R_i^{j_n(i)}}, \quad \lambda_c^{n+1} = \frac{D_i^{j_n(i)} - D_i^{j_{n+1}(i)}}{C_i^{j_{n+1}(i)} - C_i^{j_n(i)}} \tag{13}$$

with $\{R_i^{j_{n+1}(i)}, D_i^{j_{n+1}(i)}, C_i^{j_{n+1}(i)}\}$ the rate, distortion and complexity estimates of the next truncation point $(j_{n+1}(i))$ and λ_r^n, λ_r^n the slope values of $j_n(i)$, defined analogous to (13).

An algorithm to determine the adaptation point for the AUs of each GOP under joint rate and complexity constraints $R_{\max}$ and $C_{\max}$ is given in Fig. 3. Unlike previous multimedia streaming approaches that considered only the network limitations, the proposed joint R-D-C optimization may result in the transmission of a bitstream at a lower rate than that provided by the channel because higher rate streams could not be appropriately decoded by the receiver, thereby wasting unnecessary channel bandwidth.

One possible solution for implementing the above R-D-C adaptation in a streaming scenario is to consider the file format of the media. We introduce an *abstraction* layer referred to as "multitrack hinting", which is an extension of the hinting mechanism that is part of the MP4 file format specification [30]. Multitrack hinting allows structuring compressed video into multiple prioritized substreams that can be independently transmitted through multiple (RTP) channels, as illustrated in Fig. 4.

Proposed R-D-C based adaptation for each GOP :

Initialization

- Let $adapt = \varnothing$ represent the set of R-D-C optimized adaptation points (adaptation points)

Metadata Generation and Transmission

- Establish the set of AUs b_i , $1 \leq i \leq N$ of the GOP
- Establish the set of bitstream adaptation points and their corresponding R-D-C metadata: $\forall b_i : j(i), \left\{ R_i^{j(i)}, D_i^{j(i)}, \{W^{op}\}_i^{j(i)} \right\}$

Adaptation Mechanism

- For the current GOP:
  Determine $R_{\max}$ (for eq. (2)) based on the network monitor
  Determine $C_{\max}$ (for eq. (11)) based on the resource monitor
- For each AU b_i , $1 \leq i \leq N$, of the GOP:
  Establish $C_i^{j(i)}$ from eq. (10) (GCMs-to-RCMs)
  Invalidate all points $j_n(i)$ for which: $\lambda_r^{n+1} > \lambda_r^n$ or $\lambda_c^{n+1} > \lambda_c^n$
- Given $R_{\max}$, find $\{j_r^*(i), \lambda_r^*\}_{\forall b_i}$ from eq. (2) (which corresponds to eq. (12) with $\lambda_c = 0$)
- Given $C_{\max}$ and $\{j_r^*(i), \lambda_r^*\}_{\forall b_i}$, find $\{j^*(i), \lambda_r^*, \lambda_c^*\}_{\forall b_i}$ from (12) with the additional constraint: $\forall i : \frac{D_i^{j_p(i)} - D_i^{j^*(i)}}{R_i^{j^*(i)} - R_i^{j_p(i)}} \geq \lambda_r^*$, where $\left\{ R_i^{j_p(i)}, D_i^{j_p(i)} \right\}$ correspond to the previous (valid) truncation point $j_p(i)$
- For each AU b_i , $1 \leq i \leq N$, of the GOP:
  set $adapt = adapt \cup \{ j^*(i) \}$

Fig. 3. Pseudocode for the optimized R-D-C adaptation.

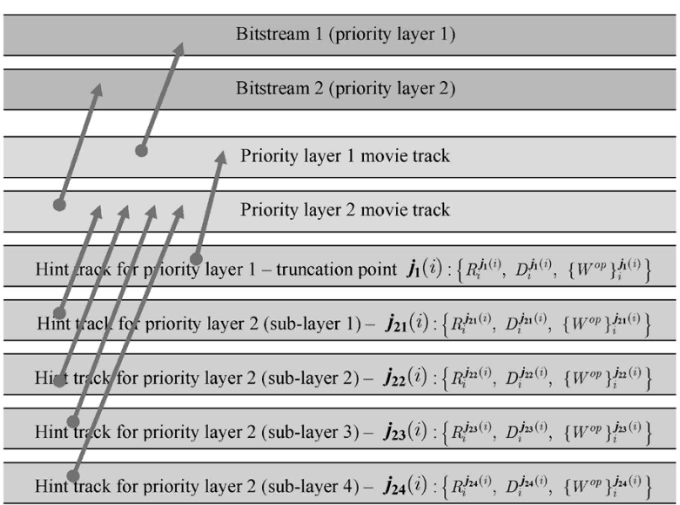


Fig. 4. Proposed multitrack R-D-C hinting file format.

This extension to conventional hinting mechanisms provide the flexibility necessary for network- and complexity-adaptive multimedia streaming by adjusting the number and type of transmitted (sub) streams. Using our multitrack hinting method, each bitstream remains unchanged and it is stored once, but it is virtually divided into multiple substreams having different corresponding rates, distortions and complexities [31]. This means that, at the AU-level, the adaptation points $j(i)$ and their corresponding R-D-C metrics $\{R_i^{j(i)}, D_i^{j(i)}, \{W_{\rm op}\}_i^{j(i)}\}$ are predetermined at the hinting stage, i.e., post encoding, but prior to the actual transmission.

TABLE I

MULTIPLICATIONS AND ADDITIONS PER PIXEL OF EACH GOP OF TWO TYPICAL SEQUENCES AT THREE DIFFERENT ADAPTATION POINTS. THE NUMBERS IN PARENTHESES PRESENT THE PREDICTION BASED ON THE PROPOSED GCM MODEL APPROACH

Sequence	Hall Monitor			Ice		
Bitrate/ frame-rate	32kbps 15Hz	64kbps 30Hz	256kbps 30Hz	256kbps 15Hz	512kbps 30Hz	2048kbps 30Hz
$op =$ $multiply$	6 (6)	14 (14)	16 (15)	48 (45)	119 (125)	130 (128)
$op =$ add	24 (30)	55 (54)	60 (65)	176 (214)	400 (394)	428 (436)

V. EXPERIMENTAL RESULTS

We first present some results that demonstrate the feasibility of the proposed model in predicting generic complexity metrics for a given decoding system. In addition, an example of the R-D-C based adaptation is presented with a prototyped multimedia streaming system.

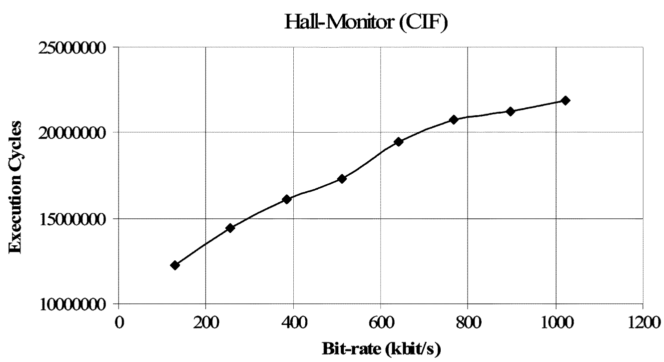
A. Validation of the Proposed GCM Estimation Model

The proposed model for the GCM estimation of video decoding was tested using the spatial-domain motion-compensated temporal filtering (SDMCTF) codec proposed recently by Andreopoulos *et al.* [25]. The chosen decoder implementation combines a number of advanced features such as adaptive temporal decomposition with long temporal filters, variable block-size multihypothesis prediction and update steps, subpixel accurate motion compensation, and JPEG2000-alike texture coding. A number of video sequences not belonging to the training set were encoded. From the compressed bitstream of each sequence, several substreams were extracted that correspond to different bitrates. In this paper we present results relating to the estimation of the computational complexity, which is quantified with the number of integer additions and multiplications per pixel. A natural extension of the presented results involves estimation of the assignment operations (memory accesses). Some preliminary results concerning the profiling of memory accesses have been presented recently by Verdicchio *et al.* [32]. We find that the effect of the memory accesses in the total decoding complexity relates strongly to a certain memory partitioning into local (internal) and nonlocal (external) accesses. This requires a specific assignment of the memory footprint of the MCTF-based decoder to a certain category of architectures and consists a topic of future research.

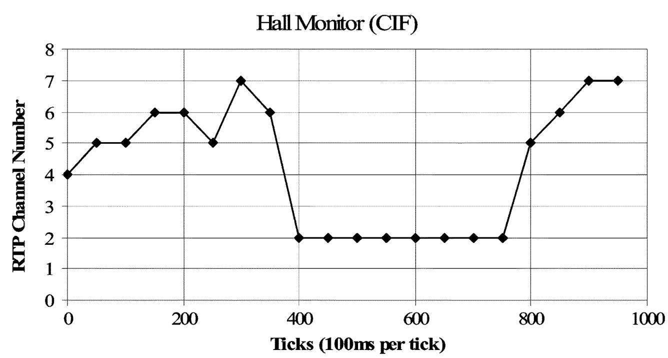
Typical experimental results for the derived GCMs per GOP (software profiling), as well as the model-based GCMs ((3)–(8) with the estimated complexity-decomposition coefficients) are illustrated in Table I. For a more detailed analysis of the decoding components with varying arithmetic complexity, we excluded the measurements for the IDWT since in the utilized codec implementation the inverse transform is performed with a fixed number of arithmetic operations per pixel. Notice that the decoder arithmetic complexity can increase by a factor of three for an eighth-fold increase in decoding bitrate. Table II presents the average model estimation-error (per GOP) for different sequences. For each sequence, the average error of the model prediction over a set of adaptation points is presented. These points represent an eighth-fold increase over the lowest (base-layer)

TABLE II
PERCENTAGE OF ERROR BETWEEN THE MODEL PREDICTION AND THE ACTUAL MEASURED ADDITIONS AND MULTIPLICATIONS PER PIXEL. FOR EACH GOP OF EVERY SEQUENCE, THE AVERAGE ERROR OVER A NUMBER OF ADAPTATION POINTS IS PRESENTED

Sequence	Average error (%) *multiply*	Average error (%) *add*
Hall Monitor	5.7	5.7
Ice	3.5	1.2
Harbour	17.6	13.4
Canoa	13.5	14.3
Total average:	**10.08**	**8.65**



(a)



(b)

Fig. 5. R-D-C driven multimedia adaptation.

bitrate. In total, the average error between the experimental and model-based GCMs was found to be approximately 10%. Note that this estimation accuracy is achieved without focusing on the specific MCTF coding algorithm or the texture-coding features of the codec. Moreover the software profiling was performed in a generic manner, without isolating specific implementation features.

B. R-D-C Driven Multimedia Streaming

To validate the proposed R-D-C adaptation, we utilize a prototyped a real-time multimedia streaming system based on the open-source MPEG4-IP software [33], [13]. The results obtained with the proposed R-D-C adaptation mechanism are illustrated in Fig. 5. In particular, Fig. 5(a) shows the average-RCM vs. bitrate results associated with eight RTP channels. The RCMs were generated from the model-based GCMs of this paper (see Table I—"Hall Monitor" sequence) with the methodology described in our recent work [7], and are expressed in terms of execution cycles in an Intel Pentium IV processor. Fig. 5(b) shows the number of channels transmitted to the receiver based on rate and/or complexity constraints. Initially, the adaptation was solely done based on network characteristics information (i.e., rate-distortion constraints) provided by the UEDs. Subsequently, the receiver started another complex application (during the time $\mathrm{tick} = 400$ until $\mathrm{tick} = 750$), and hence, the complexity of the transmitted bitstream was adapted based on the receiver RCMs.

We determined that the storage overhead associated with multi-track R-D-C hinting is about 5%–15% of the total bitstream, depending on the packet size chosen for a particular hinting scheme [12]. Moreover, we found that increasing the number of hint tracks (Fig. 4) only causes negligible increase in overhead due to the fact that, the amount of information generated by hinting is determined by the total number of hinted packets; when the packet size is selected, the total number of packets is also consequently determined independent on whether they are hinted using one track or multiple tracks. Alternatively, if the GCMs are sent to the receiver, the transmission overhead is low (less than 5%): only one number for each op is transmitted per AU.

VI. CONCLUSIONS AND FUTURE WORK

The recently standardized MPEG-21 framework enables adaptation based on both terminal capabilities and network conditions. We propose a new and generic R-D-C model that can generate metadata necessary for MPEG-21 DIA based on the available receiver resources. We illustrate the simplicity and accuracy of the proposed complexity model by predicting the computational complexity of motion-compensated wavelet video coders. The average error between the experimental and model-based generic complexity metrics was found to be approximately 10%. The proposed network and receiver R-D-C adaptation was also validated using a real-time multimedia streaming test-bed. The derived complexity model is generic and can also be used for other video coders. As a result, we believe that the proposed complexity-related descriptions could be useful for standardization purposes. In our future research, we plan to quantify the extent to which continuous adaptation to the receiver resource complexity can improve the end-to-end performance of multimedia delivery and the battery life of receivers. Another aspect of our future research is investigating the accuracy of different mappings between GCMs and RCMs for various architectures.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editors, Prof. F. Pereira and Dr. A.Vetro, for their valuable comments.

REFERENCES

[1] M. T. Sun and A. R. Reibman, Eds., *Compressed Video Over Networks*. New York, NY: Marcel Dekker, 2001.
[2] *ISO/IEC 21 000-7 FDIS Part 7: ISO/IEC JTC 1/SC 29/WG 11/N6168*, Dec. 2003.
[3] A. Vetro and C. Trimmerer, "Digital item adaptation: Overview of standardization and research activities," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 418–426, Jun. 2005.

[4] S. Saponara, K. Denolf, G. Lafruit, C. Blanch, and J. Bormans, "Performance and complexity co-evaluation of the advanced video coding standard for cost-effective multimedia communications," in *EURASIP J. Appl. Signal. Process.*, Feb. 2004, pp. 220–235.
[5] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 704–716, Jul. 2003.
[6] M. Ravasi, M. Mattavelli, P. Schumacher, and R. Turney, "High-level algorithmic complexity analysis for the implementation of a motion-JPEG2000 encoder," in *Workshop on Power and Timing Mod., Opt. and Simul., PATMOS 2003*, Torino, Italy, Sep. 2003, pp. 440–450.
[7] G. Landge, M. van der Schaar, and V. Akella, "Complexity metric driven energy optimization framework for implementing MPEG-21 scalable video decoders," in *Proc. IEEE Int. Conf. Accoustics, Speech, and Signal Processing (ICASSP-05)*, Philadelphia, PA, Apr. 2005, to be published.
[8] M. van der Schaar and P. H. N. de With, "Near-lossless complexity-scalable embedded compression algorithm for cost reduction in DTV receivers," *IEEE Trans. Consumer Electron.*, vol. 46, no. 4, pp. 923–933, Nov. 2000.
[9] F. Argenti, F. Del Taglia, and E. Del Re, "Audio decoding with frequency and complexity scalability," *Proc. Inst. Elect. Eng., Vis., Image, Signal Process.*, vol. 149, no. 3, pp. 152–158, Jun. 2002.
[10] W. Pan and A. Ortega, "Complexity-scalable transform coding using variable complexity algorithms," in *Proc. IEEE Data Compression Conf. (DCC-00)*, Mar. 2000, pp. 263–272.
[11] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Select. Areas Commun.*, vol. 15, no. 6, pp. 983–1001, Aug. 1997.
[12] M. van der Schaar, Y. Andreopoulos, and Q. Li, "Real-time ubiquitous multimedia streaming using rate-distortion-complexity models," in *IEEE Global Telecom. Conf. (GLOBECOM-04)*, vol. 2, Dec. 2004, pp. 639–643.
[13] R. Chen, M. van der Schaar, and Q. Li, "Complexity-adaptive streaming architectures for video multicasting to CE devices," in *Proc. IEEE Int. Conf. Consumer Electronics (ICCE-03)*, Jul. 2003, pp. 132–133.
[14] M. van der Schaar, D. Turaga, and V. Akella, "Rate-distortion-complexity adaptive video compression and streaming," in *IEEE Int. Conf. Image Processing (ICIP-04)*.
[15] M. van der Schaar, "Targeted Scalable Multicast Based on Client Bandwidth or Capability," http://www.uspto.gov/patft/index.html, Jan. 17, 2002. filled.
[16] J. Valentim, P. Nunes, and F. Pereira, "Evaluating MPEG-4 video decoding complexity for an alternative video verifier complexity model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 1034–1044, Nov. 2002.
[17] *Information Technology—Coding of Audiovisual Objects—Part 2: Visual*, Dec. 1999.
[18] D. Sow and A. Eleftheriadis, "Complexity distortion theory," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 604–608, Mar. 2003.
[19] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd ed. San Fransisco, CA: Morgan Kaufmann, 1995.
[20] D. Mukerjee, E. Delfosse, J.-G. Kim, and Y. Wang, "Terminal and network quality of service," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 454–462, Jun. 2005.
[21] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.
[22] T. Wiegand, Ed., "Joint final committee draft (JFCD) of joint video specification (ITU-T Rec. H.264, ISO/IEC 14 496-10 AVC)," in *Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-D157* Klagenfurt, Germany, Jul. 2002.
[23] J. Barbarien, A. Munteanu, F. Verdicchio, Y. Andreopoulos, J. Cornelis, and P. Schelkens, "Motion and texture rate-allocation for prediction-based scalable motion-vector coding," *Signal Process.: Image Commun.*, no. 4, pp. 315–342, Apr. 2005.
[24] D. Taubman and M. Marcellin, Eds., *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.
[25] Y. Andreopoulos *et al.*, "In-band motion compensated temporal filtering," *Signal Process.: Image Commun.*, vol. 19, no. 7, pp. 653–673, Aug. 2004.
[26] ATOMIUM [Online]. Available: http://www.imec.be/atomium
[27] D. H. Albonesi *et al.*, "Dynamically tuning processor resources with adaptive processing," *IEEE Computer*, vol. 36, no. 12, pp. 49–58, Dec. 2003.
[28] D. Narayanan and M. Satyanarayan, "Predictive resource management for wearable computing," in *Proc. Int. Conf. Mobile Systems, Applications, and Services*. San Francisco, CA, May 2003.
[29] J. Reichel, "Predicting the complexity of signal processing algorithms," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2001)*, vol. 3, Thessaloniki, Greece, Oct. 2001, pp. 318–321.
[30] D. Singer, W. Belknap, and G. Franceschini, ISO media file format specification—MP4 technology under consideration for ISO/IEC 14 496-1:2002 Amd 3, in Committee Draft, ISO/IEC JTC1/SC29/WG11 MPEG01/N4270-1, Jul. 2001.
[31] Q. Li and M. van der Schaar, "A flexible streaming architecture for efficient scalable coded video transmission over IP networks," in *ISO/IEC JTC 1/SC 29/WG 11/M8944*, Oct. 2002.
[32] F. Verdicchio *et al.*, "Scalable video coding based on motion-compensated temporal filtering: complexity and functionality analysis," in *Proc. IEEE Int. Conf. Image Processing (ICIP 2004)*, Singapore, Oct. 2004, to be published.
[33] MPEG4IP: Open Source, Open Standards, Open Streaming [Online]. Available: http://mpeg4ip.net

Mihaela van der Schaar (SM'04) received the M.Sc. and Ph.D. degrees in electrical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands.

She is currently an Assistant Professor in the Electrical and Computer Engineering Department, University of California, Davis. Between 1996 and June 2003, she was a Senior Member of Research Staff at Philips Research, both in The Netherlands and the U.S., where she led a team of researchers working on scalable video coding, networking, and streaming algorithms and architectures. From January to September 2003, she was also an Adjunct Assistant Professor at Columbia University, New York. Since 1999, she has been an active participant to the MPEG-4 standard, for which she received an ISO recognition award. She is currently chairing the MPEG Ad-Hoc group on Scalable Video Coding, and is also Co-Chairing the Ad-Hoc group on Multimedia Test-bed. She has co-authored more than 90 book chapters, conference and journal papers in this field and holds 11 patents and several more pending.

Dr. van der Schaar is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and of the *SPIE Electronic Imaging Journal*, and a Guest Editor of the *EURASIP* Special Issue on Multimedia over IP and Wireless Networks. She was elected Member of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society. She has also chaired and organized many conference sessions in this area and was the General Chair of the Picture Coding Symposium 2004.

Yiannis Andreopoulos (M'01) received the electrical engineering diploma and the M.Sc. degree in signal and image-processing systems from the University of Patras, Patras, Greece. He is currently pursuing the Ph.D. degree at the Vrije Universiteit Brussel, Brussels, Belgium.

His research interests are in the fields of transforms, fast algorithms, and video coding. He has made several technical contributions to the ISO/IEC JTC1/SC29/WG11 (MPEG) committee (Interframe Wavelet Exploration Activity, Scalable Video Coding).