RATE-DISTORTION-COMPLEXITY ADAPTIVE VIDEO COMPRESSION AND STREAMING

Mihaela van der Schaar*, Deepak Turaga+, Venkatesh Akella*
*Department of Electrical & Computer Engineering
University of California, Davis
+ Sony Electronics, San Jose, USA

ABSTRACT

In this paper, we discuss the benefits of complexity-driven streaming and define a realistic Rate-Complexity-Distortion framework that can assist multimedia streaming systems. We show an illustrative example for the modeling of the decoding complexity of state-of-the-art motion-compensated wavelet coding schemes. We also present a concrete system that uses these complexity metrics, which can help us quantify the performance that can be gained for multimedia streaming applications by considering complexity constraints and architectural features. Finally, we show results obtained using the proposed complexity-driven streaming architecture.

1 INTRODUCTION

Traditionally, the problem of multimedia compression and streaming has been studied within the rate-distortion theory framework that defines the trade-offs between information rate and distortion [9]. Scalable video coding, distortion-optimized packet scheduling, network-adaptive source/channel coding are examples of such research. In our approach, we are interested in studying the relationship between multimedia streaming applications and their implementation complexity at the receiver. By complexity we mean the time complexity i.e. amount of time taken to execute a function on a given receiver. This depends on the architecture of the receiver and the amount of resources available at any given time. Resources include hardware resources such as memory, processors, functional units in the processors, the instruction set, co-processors, instruction and data bandwidth and of course amount of energy (battery life) available. Resource constraints at the receiver can be dynamically changing depending on what is executing on the processor at a given time. Our goal is to investigate techniques to adapt multimedia transmission and reception to the resources available with the receiver.

The novelty of our approach is in *virtualizing complexity* i.e, we explicitly model the architecture and the resources available at the receiver as a complexity metric and develop complexity-scalable algorithms at the application layer for decompression, channel decoding, error handling etc. The receivers can choose to operate at the *desired* complexity level based on their resource constraints by negotiating with the media server the transmission of a particular complexity bitstream. Note that a similar approach could also be adopted at the server or proxy side, where the application-layer algorithms can be adapted based on the available resources.

Another merit of our approach is that our analysis and proposed framework are not based on the worst-case complexity measurements as employed in most related research, but rather we adopt "average" decoding and streaming complexities computed through a stochastic approach that considers the transmission bit-rate and video sequence characteristics.

This paper is organized as follows. We briefly discuss the concept of complexity scalability in Section 2. In Section 3, we illustrate several mechanisms for modeling the rate-distortion-complexity (R-D-C) of video. Section 4 presents a receiver driven architecture that can take advantage of the R-D-C models. We also present a brief result using the proposed complexity driven streaming architecture. We conclude with a summary in Section 5.

2 R-D-C SCALABILITY

Multimedia coding and streaming can be implemented using a diverse set of algorithms resulting in different complexities at the receiver-side. Moreover, each such algorithm can be implemented at different *"profile levels"*. For instance, an inverse discrete wavelet transform (IDWT) can be implemented at various profile levels corresponding to different accuracy-complexity trade-offs using e.g. floating-point, integer, multiplication-free arithmetic. Another example of different profiles is a motion-compensated wavelet coding scheme like in [1], which can employ different spatio-temporal decomposition levels. In this paper, the term "complexity profile" will be used to determine the upper bound of the complexity implementation for an algorithm. Most existing literature aims at quantifying such upper bounds for the complexity of an algorithm [3][4]. While measuring such upper bounds is extremely important for determining e.g. ASICs costs, it is not very meaningful for quantifying the complexity of implementing an algorithm on programmable, reconfigurable or energy-adjustable processors.

Consequently, in our paper we will employ *average* complexity metrics for an algorithm at a specific profile level, which will deviate significantly from the maximum complexity value for most algorithms depending on the employed transmission bit-rate and multimedia data characteristics

Within the proposed framework, the variation of the complexity of a specific algorithm (with a given fidelity) as a function of the bit-rate and distortion is denominated *complexity scalability* and can be defined as a transition between two operating points with different complexity coordinates in the achievable region. Specifically, in this paper, we are interested in determining useful models for the complexity of the various decompression algorithms that can be easily employed in our proposed resource/receiver-dependent streaming.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

3 COMPLEXITY MODELS

In this section we will describe our approach to model the complexity of a receiver as well as a framework for building these models and metrics.

3.1 Complexity Modeling Framework

We introduce the notion of available computational resources of a receiver, which is a function of the receiver's hardware/software architecture and resource constraints, etc. Note that the available computational resource of a receiver varies over time, because it could depend on the current workload and its priority, the available energy (battery life), amount of memory available, and the contents of the memory etc.

In our framework, the server can employ various algorithms leading to different computational complexities at the receiver side and hence, the receivers can subscribe to appropriate bitstreams based on their available computational resources. For this, a model for the complexity of the algorithms that considers generic architecture parameters should be determined at the server side. Given the number of factors that influence the complexity of the receiver, it is impractical to determine at the server side the *specific* complexity for all possible target receiver architectures. Consequently, we propose to adopt an abstract *Generic Complexity Model* that captures the abstract/generic complexity of the employed decoding or streaming algorithm depending on the multimedia data characteristics and transmission bit-rate (or any other network characteristics parameter that affects the decoding or streaming process, e.g. packet-loss, probability of path failure etc.). Subsequently, we refer to the complexity measures determined using the Generic Complexity Model as *generic complexity metrics (GCMs).* The GCM can include the number of additions/subtractions/shifts, or multiplications /divisions, memory accesses or more sophisticated operations. Note that the GCMs necessary for the decompression and streaming can be transmitted to the receiver at different granularities (e.g. for each functional unit, sub-unit etc.) and for different sizes multimedia processing units (MU). MUs can be a group of video blocks, a video frame, a group of frames (GOP) etc. The advantage of identifying and transmitting GCMs at finer granularities is that the receiver can request (see Figure 5 for a system diagram) a bitstream generated by a specific set of decoding functional sub-units, and delivered by a streaming algorithm having a specific set of functional sub-units that best fit its architecture and resource constraints. A finer granularity will enable more accurate adaptation by enabling the receiver to choose specific algorithm sub-units at different profiles characterized by different operational Rate-Distortion-Complexity (R-D-C) surfaces and corresponding GCMs, at the expense of an increased communication overhead between the server and receiver and possibly an increased computational overhead at the server side associated with generating GCMs for a higher number of functional sub-units. However, it should be noted that employing finer GCMs granularities is only beneficial if a set of alternative bitstreams (resulting in different decoding complexities) and streaming algorithms can be used at the server side. Otherwise, the receiver cannot benefit from the increased granularity GCMs and bandwidth is unnecessarily wasted for their transmission.

The MU size can be changed depending on the application requirements (e.g. tolerable delay), desired granularity of complexity adaptation, etc. Determining optimal GCMs granularities and MU sizes for different algorithms, receiver architectures and how they affect the R-D-C performances of different multimedia application classes forms an important part of our research.

The GCMs for the decoding and streaming algorithms will be communicated to the receiver, which will then translate them using a *Generic Risc/Reference Machine (GRM)* into an architecture-specific complexity measure called *real complexity metrics (or RCM).* Based on the available complexity resources and the RCMs, receivers will request from the server appropriate bitstreams that can be decoded and streamed given the receivers' resources and energy efficiency constraints.

3.2 GCM Modeling Example

In this section we illustrate how a GCM can be built for a motion-compensated wavelet video coding algorithm based on the UMCTF [2]. We measure the decoder complexity (using one frame as a MU) in terms of the number of non-zero coefficients before IDWT, as this determines the number of additions and multiplications. We then build a simple abstract model for the quality and complexity profiles when decoding at different bit-rates. The proposed model depends on the underlying content characteristics and corresponding encoding parameters. We consider a specific decomposition scheme with three temporal decomposition levels before the spatial transform and a GOP of 8 frames. The decoder complexity and PSNR of each frame in the GOP for the Coastguard sequence at different bit rates is shown in Figure 1.

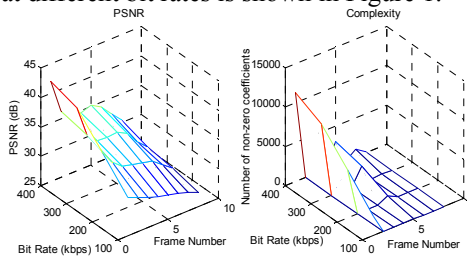

Figure 1. R-D and complexity performance for Coastguard

To construct the model, we examine the relationship between the decoded PSNR and the bit-rate and the complexity and bit-rate for each MU. We show the mean number of non-zero coefficients for different frames (averaged across four GOPs) in Figure 2.

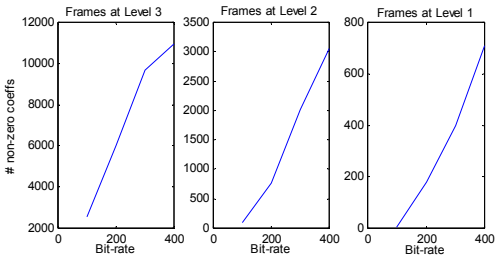

Figure 2. Mean complexities averaged across 4 GOPs

We show these numbers for frames that are temporally high-pass filtered at different levels. We do not show any frames from the highest level in the temporal pyramid (temporal level 0), since they do not contribute significantly to the decoding complexity (most coefficients are 0, due to good temporal filtering from nearby frames). The mean complexity m_C shows an approximately linear relationship with the bit-rate *r*, and may be modeled as $m_C = \alpha r + \beta$, where (α, β) are the model parameters. The trained parameters, using frames from two GOPs as training set, are (28.8, 82.5), (10.1, -1040) and (2.35, -250) for frames at levels 3, 2 and 1 respectively. The variance in complexities for each of these different levels is small. Hence, the complexity corresponding to this chosen decomposition scheme may be modeled as a set of iid Gaussian random variables with the mean a linear function of the bit-rate and a fixed small variance.

The R-D performance of motion-compensated wavelet coding schemes resembles that of hybrid coding schemes [1], which is known to have approximately log-linear relationship with the bit-rate [5]. Using these models, we can create the R-D and complexity profiles for the Coastguard sequence at different bit-rates, corresponding to the chosen decomposition scheme. We show an example in Figure 3.

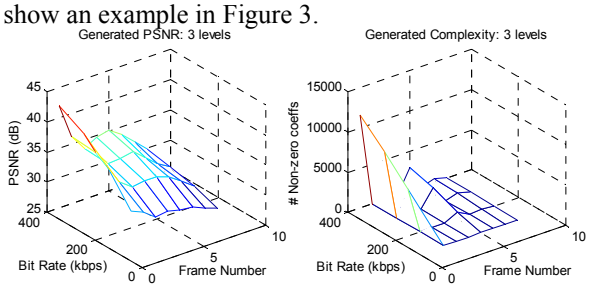

Figure 3. R-D and Complexity profiles generated using model

These models accurately (within ±10%) predict the decoded PSNR and complexity for the test set consisting of frames from the remaining two GOPs. The decoder can then use these models to decide on the decoding bit-rate based on its complexity constraints and quality requirements. The models we chose are very simple and correspond to very specific content and decomposition structures. More detailed models can be created, similar to work in bitstream modeling [5][6][7] to improve their accuracy and make them more general.

4 RECEIVER COMPLEXITY-DRIVEN MULTIMEDIA STREAMING

In this section we illustrate how receivers can use these complexity metrics and subscribe to different bitstreams requiring different decoding algorithms or different streaming algorithms based on their resource constraints. In our proposed multimedia streaming framework, we rely on an *abstraction* layer referred to as "multi-layer hinting", which is an extension of the standardized hinting mechanism of [8]. Multi-track hinting allows structuring compressed video into multiple prioritized sub-streams (layers) that are transmitted through independent channels (e.g. RTP channels). This multi-to-one relationship between hint tracks and a movie track (see Figure 4 for an illustration of the multi-hinting concept) breaks the currently used one-to-one relationship employed for hinting compressed video [8], and provides the flexibility necessary for network- and complexity-adaptive multimedia streaming by adjusting the number and type of transmission channels. Using our multi-track hinting method, each bitstream remains unchanged and it is stored once, but it is virtually divided into multiple sub-streams. The hinting algorithm will allocate a specific bit-rate and GCMs to each hint track. Consequently, the bit-rate and GCMs for each channel (i.e. RTP connection) is predetermined at the hinting stage (i.e. post encoding but prior to the actual transmission). The multi-track hinting method can also be extended to include "on-demand" error protection channels (such as Forward Error Correction (FEC), retransmission etc.), adaptive packetization strategies, complexity scalability features etc.

Figure 4. Proposed multi-track R-D-C hinting file format.

We also introduce the concept of "channel profile" for which specific video layers and FEC protection streams are grouped together to form optimized transmission profiles characterized by specific GCMs and suitable for specific network characteristics . For the same video, multiple channel profiles may be available for the receivers. The priority of the various channels within such a profile is pre-determined due to channel dependency and allows for easy adaptation by the server. Note that small extensions to the existing Internet streaming protocols (RTSP, SDP) are necessary for the realization of the proposed framework, but they are not discussed here.

Note that the proposed receiver complexity-driven streaming framework can be applied in conjunction to any compression scheme (e.g. Data Partitioning as employed in H.264 and MPEG-4 standards, prioritized transmission of different frame types – I,P,B, wavelet based spatial, SNR or FGS scalabilities or equal importance layers like in Multiple Description Coding etc.).

We illustrate our proposed receiver complexity-driven streaming strategy with a simple real-time video streaming example, consisting of the following steps:

STEP 1: The raw video captured by a camera is compressed using several encoders or encoder settings (e.g. using different accuracies for the motion estimation and thus compensation at the decoder side). In this illustrative example, two different bitstreams are compressed using different encoders. Also, different protection tracks (e.g. FEC or retransmission) are generated.

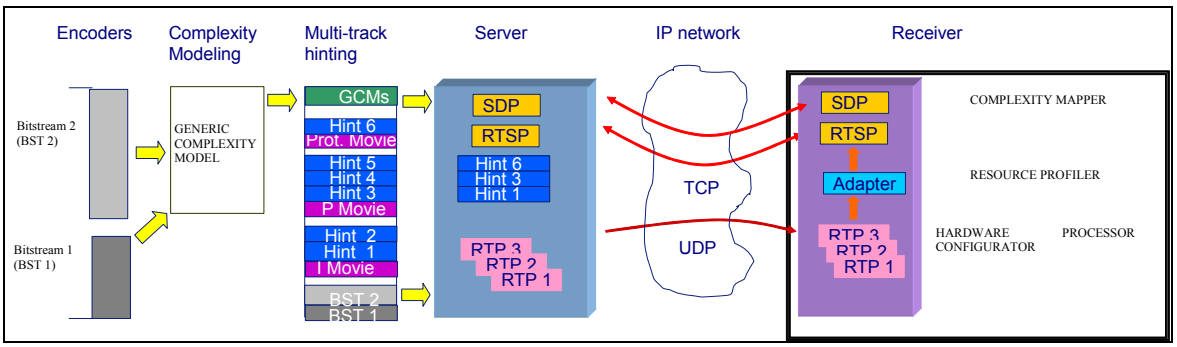

Figure 5. Receiver Complexity-Driven Streaming System

STEP 2: The generic decoder and streaming algorithms (employing different protection strategies) complexities (i.e. GCMs) for the different bitstreams are computed using *a priori* determined GCMs.

STEP 3: Hinting tracks and their corresponding bit-rates and GCMs are determined on-the-fly or *a priori* and subsequently are stored at the server side together with the bitstreams.

STEP 4: The media server and the client in the receiver set up a streaming session through network protocols such as RTSP, RTP and SDP, the key enhancement being the usage of multi-track hinting and communication of the different complexity profiles (characterized by GCMs) to the receiver. The server offers multiple channel classes/profiles? options that the receivers can choose (subscribe to) based on network conditions or generic complexities metrics GCMs. The different channel profiles and the corresponding GCMs are transmitted to the receiver using SDP.

STEP 5: The receiver has a new module called COMPLEXITY MAPPER, which translates the GCMs of the bitstreams and streaming algorithms available at the server-side into architecture specific complexity metrics, (i.e. RCMs). Based on the computed RCMs and the available receiver resources and network characteristics, the receiver will subscribe to different bitstreams using RTSP. The module called RESOURCE PROFILER inside the receiver monitors the various resources such as caches, FIFOs, functional unit utilization, power dissipation etc. and chooses the appropriate complexity level for the receiver. User preferences can also be easily incorporated in this framework.

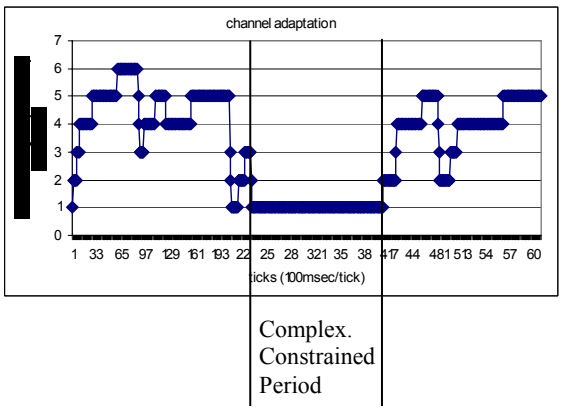

Figure 6. R-D-C driven streaming adaptation

The responsiveness of complexity-based channel adaptation is determined by the adaptation algorithm involved at the receiver-side. Figure 6 illustrates the performance that can be obtained using our proposed R-D-C streaming framework. Initially, the adaptation was solely done based on bandwidth (R-D) constraints. Subsequently, the receiver started another complex application (during the time tick=23 until tick=41), and hence, a lower complexity stream was requested from the server to meet this receiver constraint.

6 CONCLUSIONS

Previous research in multimedia streaming focused mainly on coping with network variations [9]. In this paper, we discuss the benefits of complexity-driven streaming and define a realistic Rate-Complexity-Distortion framework that can assist multimedia streaming systems.

REFERENCES

1. J.R. Ohm, M. van der Schaar, J.W. Woods, "Interframe Wavelet Coding – Motion Picture Representation for Universal Scalability", (invited paper), *Image Communication Journal: Special Issue on Digital Cinema*, 2nd Quarter, 2004.

2. M. van der Schaar and D. S. Turaga, "Unconstrained Motion Compensated Temporal Filtering (UMCTF) for Wavelet Video Coding", Proc. ICIP, 2003.

3. Y. Chen and Z. Zhong, "Picture-wise resource allocation in MPEG-2 decoding", *Proc. ICIP*, 2001.

4. Vetro, "A minimum drift 3-Layer scalable DTV decoder", Digest of technical papers – *Proc. ICCE*, pp. 54-55, June 1998.

5. K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of Video Transmission over Lossy Channels", IEEE JSAC, Vol. 18, No. 6, June 2000.

6. K. Chandra and A.R. Reibman, "Modeling One- and Two-Layer Variable Bit Rate Video", IEEE/ACM Trans. Networking, Vol. 7, No. 3, p398-413, June 1999.

7. Z. He, and S.K. Mitra, "A Unified Rate-Distortion Analysis Framework for Transform Coding", IEEE Trans. On. Circ. And Sys. For Video Tech., Vol. 11, No. 12, December 2001.

8. D. Singer, W. Belknap, G. Franceschini, "ISO Media File Format Specification — MP4 Technology under consideration for ISO/IEC 14496-1:2002 Amd 3," Committee Draft, ISO/IEC JTC1/SC29/WG11 MPEG01/N4270-1, July 2001.

9. S. McCanne, V. Jackobson, and M. Vetterli, "Receiver-driven Layered Multicast," Proc. SIGCOMM'96, Standford, CA, Aug. 1996, pp. 117-30.