A Generic Video Adaptation Framework Towards Content - and Context - Awareness in Future Networks

Willy Aubry, Daniel Negru, Bertrand Le Gal, Simon Desfarges, Dominique Dallet

A GENERIC VIDEO ADAPTATION FRAMEWORK TOWARDS CONTENT- AND CONTEXT-AWARENESS IN FUTURE NETWORKS

Willy Aubry[1,2,3], Daniel Négru[2], Bertrand Le Gal[1], Simon Desfarges[2], Dominique Dallet[1]
waubry@viotech.net, negru@labri.fr, legal@ims-bordeaux.fr, desfarge@labri.fr, dallet@ims-bordeaux.fr

[1]University of Bordeaux,
IMS Laboratory,
CNRS UMR 5218, IPB,
Talence, France

[2]University of Bordeaux,
LaBRI Laboratory,
CNRS UMR 5800, IPB
Talence, France

[3]Viotech Communications
Montigny-le-Bretonneux, France

ABSTRACT

Network researches are more and more turned toward content and context aware features. Thus, being able to manipulate data flows and to adapt those to given constraints with a minimum resource involvement is a vast research issue. On top of those topic researches resides video manipulation. But, developing heterogeneous video transcoder takes tremendous time. Many solutions have been proposed to reduce resource consumption at runtime but involve re-development of every codec for every situation, multiplying development cost under time to market constraint. In this paper, we propose a generic framework that enables the reuse of already developed and ready to use codecs, saving time to market for next generation network devices.

Index Terms— Video adaptation, heterogeneous transcoder, generic framework

1. INTRODUCTION

In today's world, video stream is one of the most consumed data flow over Internet, with the most bandwidth demanding. Hence, video streams have the main impact on the global network. Thus, trying to transport an adapted video stream to the network characteristics has been deeply studied [9][10]. Nowadays, network oriented researches are directed toward the end user's quality of experience. The network state is not the only parameter considered for video adaptation any more. The user context, especially its terminal characteristics such as supported codec and screen resolution, is now the main constraint that has to be taken into account.

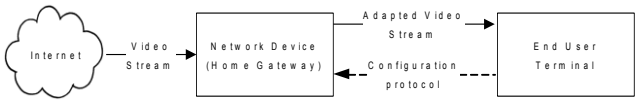

Figure 1: Generic Network Adaptation Framework

This problematic is one of the main focus in media centric ecosystem and a key to content and context awareness for next generation network. We proposed to address this problematic by using a network device as shown in Figure 1**.** The main objective is to embed a video adaptation processing engine in an external device that possesses monitoring capabilities. Then, this device will be able to detect and adapt the video contents depending on the user's context (network load, terminal used …), making it into a content/context aware network device.

This system offers a video distribution that is seamless for both the consumer and the provider. Indeed, the consumer can access video stream based only on its content without worrying on its own capability to read it. Moreover, the content provider does not have to take into account context parameters when asked for a video stream. This feature is achieved by embedding video adaptation capabilities in network devices.

For scalability purpose, the platform will be implemented in last hop devices that possess a better and quicker knowledge of the end user context. However, those devices are mainly gateways with network switching responsibilities (such as devices located along with 3G antennas or home gateways) having low-computation performances. This explains why real-time video adaptation solution[1] proposed in this paper was developed under low-computation complexity and low-cost constraints.

This article is organized as followed. In Section 2, the system design will be detailed. Heterogeneous video adaptation will arouse as the main issue of this system. A state of the art of already existing solution will be presented and a lack of generic hardware solution will be pointed out. In Section 3, our framework that overcomes this issue will be presented before to show a design example in Section 4. Conclusion and future works will be drawn in Section 5.

[1] Work supported by the French project ARDMAHN within French National ARPEGE ANR Program http://www.ardmahn.org and by the European project ALICANTE within EU FP7 ICT, under grant agreement n° 248652. http://www.ict-alicante.eu.

2. VIDEO ADAPTATION SYSTEM

2.1. System design

In the proposed system presented in Figure 2, the gateway, located between the video source coming from the Internet and the device embedded in the end user terminal, transcodes the video stream.

Adaptation is activated and configured according to the embedded device decoding capabilities. A communication protocol is used by the end user terminal to provide the list of its supported standard. This system requires two major evolutions for current systems:

1. To adapt the video characteristics, a modified *home gateway* is required. This resource that links the embedded device to the Internet (or another video provider), needs real-time video adaptation capacity. In the proposed approach, this task is implemented using a hardware accelerator (FPGA).
2. To enable and control the video adaptation process, the *embedded device* must be able to inform the modified home gateway of the supported standard and characteristics (mainly screen size). This feature is implemented using a negotiation protocol.

2.2. Real-time video heterogeneous transcoding

Video adaptation is a complex task requiring high-performance resources. However, these resources (like DSP processors) are expensive and not available in home-gateway products. Many approaches described in the literature have been proposed to reduce computation complexity of video adaptation while keeping a high video quality [1][2].

Video stream adaptation has been studied in literature to adapt video content according to the embedded devices characteristics (display size [3]-[5], implemented video codecs [6]-[8], etc.), and to reduce network load [9]-[11] (avoiding network packet drops). In a previous work [12] we studied the impact of the video spatial resolution on the power consumption of the decoding process at the terminal end.

Most of the available devices (TV, Smartphone …) have dedicated video decoding chips. Thus, the terminal device supports only a limited list of standards. Hence, it is mandatory to be able to deliver the wanted video into a proper standard while also being able to adapt its feature (bitrate, frame resolution). A codec comparison [13] showing the different characteristic of usual (Table 1) outline this fact.

Features	H.261	MPEG-1	MPEG-2	H.263	MPEG-4	H.264
Picture Coding Type	I, P	I, P, B	I, P, B	I, P, B	I, P, B	I, P, B
Entropy Coding	VLC	VLC	VLC	VLC, SAC	VLC	UVLC, CAVLC, CABAC
MV Resolution	Int. Pel	½ pel	½ pel	½ pel	¼ pel	¼ pel
Transform	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	4×4 & 8×8 Integer
Vector Block Size	16×16	16×16	16×16, 16×8	16×16, 8×8	16×16, 8×8	16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4
Spatial Intra Prediction	No	No	No	No	No	Yes
Formats Supported	Prog.	Prog.	Prog./Intr.	Prog.	Prog./Intr.	Prog/Intr
Prediction Modes	Frame	Frame	Field & Frame	Frame	Field & Frame	Field & Frame
De-Blocking Filter	In-loop	None	Post	Annex J In-loop	Post	In-loop

Table 1: CODEC comparison

Toward this issue, heterogeneous transcoders have been proposed. An h.263 to h.264 pixel domain frame transcoder has been proposed [14]. This transcoder can be used to solve the MPEG-2 to h.264 issue [8]. The MPEG-2 to h.263 problematic has also been addressed [6].

Adaptation – i.e. changing video parameters such as quantizer scale, frame resolution … – is almost always separated from transcoding – i.e. changing video CODEC. A downscaling process for h.26x (x = 1, 2 or 3) [4] and a rate control system for MPEG-2 to MPEG-4 transcoding [15] has been proposed. Adaptation has been addressed either as an homogeneous transcoding process (same CODEC) or as a specific heterogeneous transcoding process using features of the addressed CODECs. To authors' knowledge, no generic adaptation has been proposed that could be used for any kind of transcoding using a uniform approach.

3. A GENERIC HARDWARE VIDEO ADAPTATION FRAMEWORK

We propose to use a video adaptation framework shown in Figure 3. This framework aims at reusing decoder and encoder previously developed by or bought to a third party. One work motivation comes from that no generic low-computation complexity approach to transcode a video exists. Let's consider n input codecs, p output codecs and q video adaptation techniques. There exists up to $n \times p \times q$ dedicated approaches for video transcoding. Implementing such number of video transcoding chain is not realistic. To

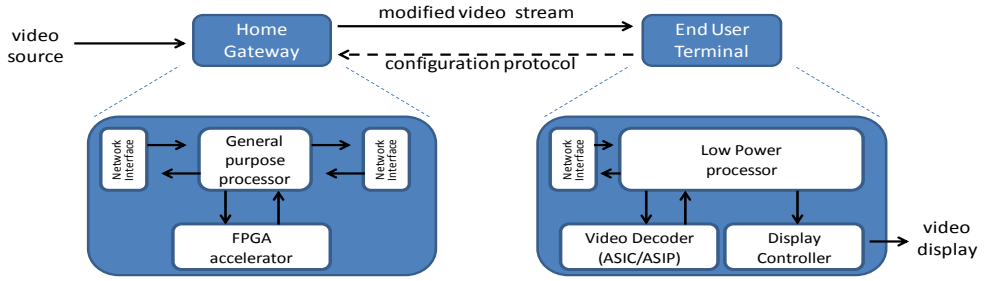

Figure 2: System Overview

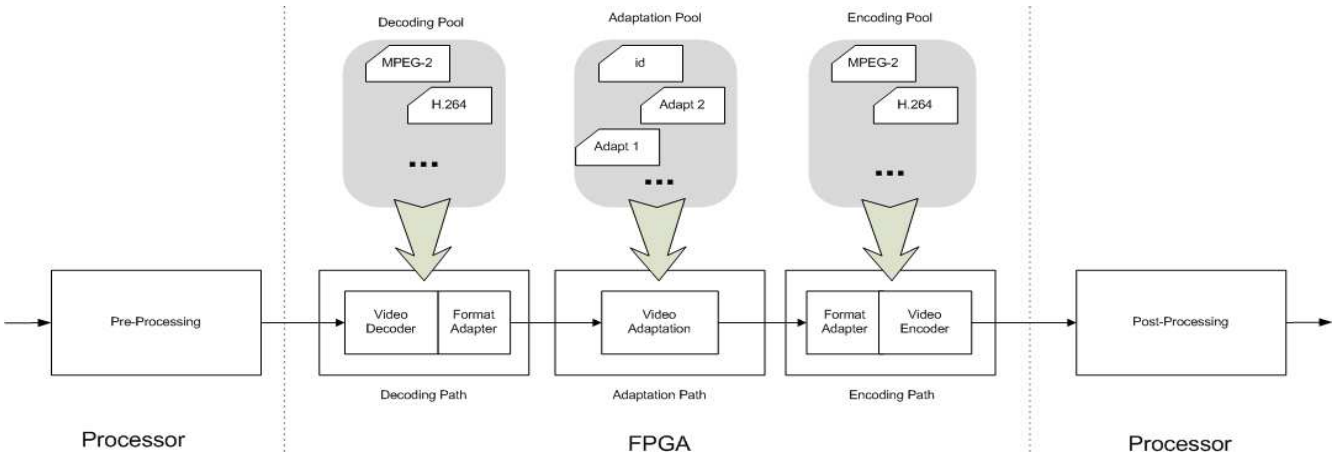

Figure 3: Adaptation Framework

reduce system development complexity we propose to divide the video adaptation processing chain in three distinct stages:

1. The first stage is dedicated to partial video decompression of the input video stream into an intermediate video format.
2. The second stage is dedicated to video parameters adaptation. This stage works on the intermediate video format to change the video characteristics (spatial resolution, framerate, quantizer scale …).
3. The third stage is dedicated to video stream reconstruction from intermediate video format.

The main concept is to have a generic process that performs video adaptation in an intermediate format and to let the heterogeneous part inside the converter. By converting to/from the intermediate format, the transcoding architecture frees the decoding and the encoding parts. Any decoder will be able to provide data for any encoder because of the format converter. This approach helps to reduce the implementation complexity through the usage of an internal video format. Indeed, implementing video adaptations from codec c_{i1} to codec c_{o1} using video parameters p_1 and p_2 only requires the implementation of hardware modules c_{i1}, c_{o1}, p_1 and p_2. This approach provides a lower complexity compared to dedicated processing developments performed according to literature ($c_{i1}+_{p1}+c_{o1}$) and ($c_{i1}+p_2+c_{o1}$) must be implemented.

Moreover, FPGA devices provide partial hardware reconfiguration opportunities during runtime [16]. This reconfiguration possibility, coupled with our proposed approach, allows short reconfiguration time to switch from a video adaptation process to another one. It means that it could support multiple users or to change during video viewing some stream characteristics by fetching the different part of the transcoding chain (decode-adapt-encode) in a pool of available design.

Once a codec has been developed with its intermediate format converter, it can be added to the pool of available codec supported by the adaptation platform. The adaptation does not need to be re-developed. The interconnection with already developed codec is seamless and saves a lot of development time as well as it adds a lot of flexibility.

3.1. The intermediate format

Intermediate format was specified according to commonly used video standard requirements. The most complex video standard is h.264. Indeed, the motion estimation is ¼ pixel, and its motion predictions are applied to pixel blocs which can have different sizes.

Using such video characteristics create the intermediate format authorize h264 to h264 video adaptation with no information loss. Moreover, other video codecs such as MPEG-2 or MPEG-4 that have lower requirements will use only a subset of the intermediate format functionalities

The intermediate format should be chosen in order to support the maximum feature of every CODEC so that the adaptation remains optimum. As shown on table 1, standards do not always use the same transform. We need a common domain to process data. The pixel domain is the obvious choice. Since h.264 has a ¼ pixel precision, the adaptation shall posses a ¼ pixel precision.

The smallest vector Block size is 4x4 for h.264 thus it will be the granularity of the generic adaptation process. With each blocks, information will be added such as motion vector, quantizer scale …

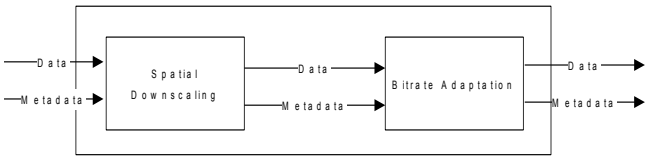

Figure 4: Generic Adaptation Process

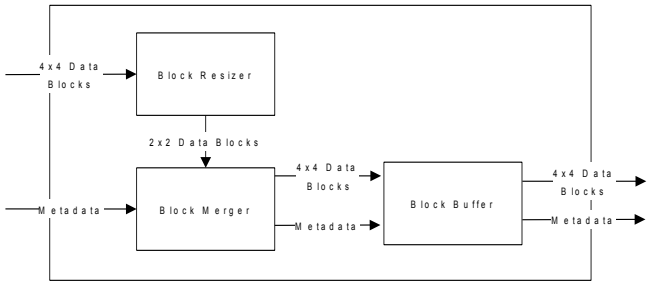

Figure 5: Spatial Downscaling Process

3.2. Generic Adaptation Process

The generic Adaptation Process (Figure 4) is composed of a spatial downscaling process and a bitrate adaptation process. Both processes are triggered according to the required adaptation and can implement almost any algorithm found in the literature.

The bitrate adaptation mainly focuses on finding the proper quantizer scale value to adjust the bitrate of the encoded video to the required bitrate. The data and metadata format do not play a role in the process and thus the bitrate adaptation process will not be tackled in this paper.

The spatial downscaling process is shown on Figure 5**.** This process is composed by a *Block Resizer* process, a *Block Merger* process and a *Block Buffer* process.

The *Block Resizer* computes the resizing (e.g. 4x4 blocks into 2x2 blocks for a ½ downsizing). The *Block Merger* merges the resized block into 4x4 blocks and merges the metadata to obtain the proper intermediate format. For a ½ downsizing example, four 2x2 blocks with their own metadata (each block his own) will become a unique 4x4 block with its metadata.

Techniques to merge information exist in the literature **[1]**. This design allows any kind of spatial downsizing techniques.

The use of *Block Buffer* is important for the case of dealing with color component. A macro-block is composed of luminance blocks and chrominance blocks in a proper order. The *Block Merger* and *Block Resizer* alter this order. The *Block Buffer* is in charge to assure this good color order.

For example, the generic downscaling process input is composed of 16 luminance blocks then 8 chrominance blocks. After the *Block Merger* the data flow will be 4 luminance blocks then 2 chrominance blocks. The role of the *Block Buffer* is to put this flow back to 16 luminance blocks then 8 chrominance blocks.

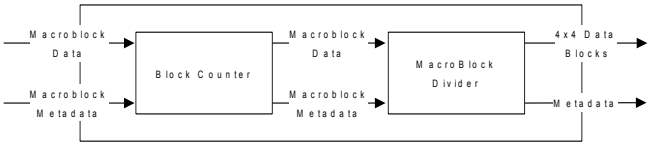

Figure 6: Decoder to Intermediate Format Converter

4. MPEG-2 FORMAT TO INTERMEDIATE FORMAT TRANSLATION

Our implementation of an MPEG-2 transcoder is made of three parts: the decoder path, the adaptation process and the encoder path.

Our decoder path gives data in 16x16 macro-blocks in YCbCr 4:2:0 format. Each macro-block is composed of 6 sub-blocks of 8x8 pixels (4 for luminance and 2 for chrominance). The information along the macro-blocks is mainly composed of Quantizer scale, motion vector and Macro-block type. The same format is used for the input of the encoder path.

4.1. Decoder format to intermediate format

The converter from the decoder format to the intermediate format is shown on Figure 6. It contains a Block Counter and a Block Divider.

A video cannot contain a non-integer number of macro-blocks per line or column. The role of the *Block Counter* is to assure by removing/adding blocks that there is an integer number of macro-blocks per line and column in the adapted frame. For instance, in a downsizing by 2, the original frame should have an even number of macro-block. If this hypothesis is not fulfilled, the *Block Counter* will remove one macro-block at the end of the line/column.

Then the converter has to split each macro-block into 4x4 blocks with the associated metadata. This is the role of the Macro-block Divider. The 4 8x8 luminance blocks are split into 16 4x4 data blocks, each of them with a copy of the metadata of the macro-block. Then the 2 8x8 chrominance blocks are split into 8 4x4 data blocks, also with a copy of the metadata as shown in Figure 7.

4.2. Intermediate format to MPEG-2 encoder format

The converter from the intermediate format to the MPEG-2 format is in charge of merging 4x4 blocks and their metadata into a 4:2:0 16x16 macro-blocks with its metadata. On the data path, the converter is reordering the data in the right order. On the metadata path decision, algorithms have to be applied such as: (a) motion vector estimation, (b) quantizer scale and (c) macroblock type. Algorithms can be found on transcoding overviews [1]. The ¼ motion vector pixel precision will be rounded to a ½ pixel precision if needed.

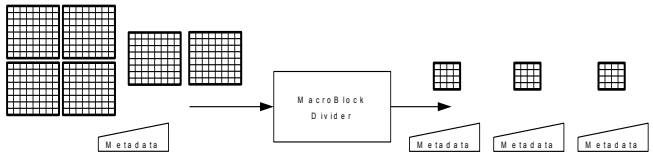

Figure 7: Macroblock Divider

4.3. Other CODEC considerations

In order to process other CODECs, the converter will almost look alike (depending on the encoder and decoder implementation). The h.264 possesses two main features which are not commonly shared with the other CODEC that will need a special care: (a) the vector block size and (b) the intra prediction.

The vector block size can be easily handled. The intermediate format grant a motion vector for each 4x4 pixel block. Hence, to compute the vector block size, a motion vector comparison with the neighbors is enough as shown in Figure . The vector block size can either be 4x4, 4x8, 8x4 or 8x8. If 8x8 blocks is created then the process runs another time in order to obtain the coarser 16x8, 8x16 and 16x16.

5. CONCLUSION AND FUTURE WORKS

In this paper, we addressed the processing chain development issues in video adaptation. These issues are mainly time to market while addressing for multi codec adaptation. We proposed a framework that understands the video adaptation process as three paths: the decoding path, the adaptation path and the encoding path. Through our framework, we propose to translate data coming out of the decoding path or coming to the encoding path into an intermediate format. This intermediate format enables the development of the three paths seamlessly which reduce greatly the time to market. Being seamless regarding the encoding and the decoding paths is the key feature for heterogeneous processing chain development.

In future works we aim at validating this framework by adding an h.264 codec in order to allow homogeneous and heterogeneous resizing for h.264 and MPEG-2. This work will allow us to tackles intermediate format issues not yet discovered. In a second time, we aim at evaluating the performance of an on board version. The cost in silicon space and throughput impact of such a framework will be evaluated. The support of a third party codec is one of the next steps in order to refine the methodology of the framework.

6. REFERENCES

[1] I. Ahmad, X. Wei, Y. S. & Zhang, Y.-Q. “Video Transcoding: An Overview of Various Techniques and Research Issues IEEE Transactions on Multimedia, October 2005.

[2] Y. Xin, C-W. Lin and M-T. Sun “Digital Video Transcoding” Proceedings of the IEEE, Vol. 93, No. 1, January 2005

[3] A. Vetro et al. “Complexity-Quality Analysis of Transcoding Architectures for Reduced Spatial Resolution” IEEE Trans. on Consumer Electronics, vol. 48, no. 3, pp. 515-521, 2002.

[4] B. Shen, I. K. Sethi andB. Vasudev, “Adaptive motion-vector resampling for compressed video downscaling” IEEE Trans. on Circuits And Systems For Video Technology, 1999.

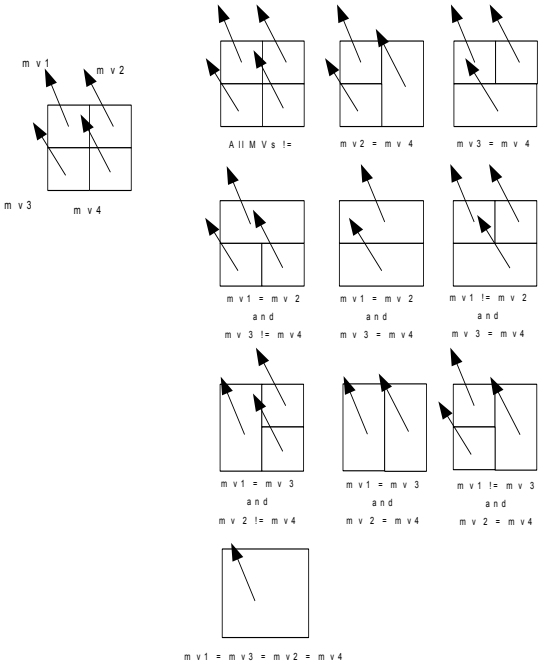

Figure 8: Vector Block Decision

[5] P. Yin et al. “Drift Compensation for Reduced Spatial Resolution Transcoding” IEEE Trans on Circuits and Systems for Video Technology, vol. 12, no. 11, pp. 1009-1020, 2002.

[6] N. Feamster and S. Wee, "An MPEG-2 to H.263 transcoder". In SPIE Voice, Video and Data Communications Conference, September 1999.

[7] J. Xin, A. Vetro and H. Sun “Converting DCT Coefficients to H.264/AVC”. IEEE Conference on Multimedia (PCM), Lecure Notes in Computer Science, Vol. 3332, pp. 939, 2004

[8] H. Kalva, B. Petljanski and B. Furht “Complexity Reduction Tools for MPEG-2 to H.264 Video Transcoding” WSEAS Transaction on Information Science & Applications, Vol. 2, pp 295-300, March 2005.

[9] Z. Lei and N.D. Georganas, "A rate adaptation transcoding scheme for real-time video transmission over wireless channels", Signal processing. Image communication, vol. 18, pp. 641-658, 2003.

[10] Eleftheriadis, A. & Anastassiou, D. Meeting “Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video”. In the 5th International Workshop on Network and Operating System Support for Digital Audio and Video, 1995

[11] M. Lavrentiev and D. Malah, "Transrating of MPEG-2 coded video via requantization with optimal trellis-based DCT coefficients modification". In Proceedings of EUSIPCO Vienna, Austria, September, 2004.

[12] W. Aubry, B. Le Gal, D. Dallet, S. Desfarges, D. Negru “A system approach for reducing power consumption of multimedia devices with a low QoE impact” ICECS 2011. 11-14 Dec. 2011.

[13] J. Golston,and A. Rao “ Video Compression: System Trade-Offs with H.264, VC-1 and Other Advanced CODECs” Texas Instruments, white paper, Aug 2006.

[14] J. Bialkowski, M. Barkowsky and A. Kaup “Overview of Low-Complexity Video Transcoding from H.263 to H.264”

IEEE International Conference on Multimedia and Expo, 2006. 9-12 July 2006.
[15] Y. Sun, X. Wei and I. Ahmad “Low-Delay rate-control in video transcoding” ISCAS 2003, 25-28 May 2003.
[16] F. Duhem, F. Muller and P. Lorenzini, "FaRM: Fast Reconfiguration Manager for Reducing Reconfiguration Time Overhead on FPGA" In proceedings of the ARC'11 Conference, Volume 6578/2011, pp. 253-260, 2011.