XML-Based Customization Along the Scalability Axes of H.264/AVC Scalable Video Coding

Davy De Schrijver*, Wesley De Neve*, Koen De Wolf*, Stijn Notebaert*, and Rik Van de Walle‡
Department of Electronics and Information Systems – Multimedia Lab
*Ghent University – IBBT
‡Ghent University – IBBT – IMEC
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
Email: davy.deschrijver@ugent.be

***Abstract*— The heterogeneity in the current and future multimedia environment requires an elegant adaptation framework for the production and consumption of different kinds of multimedia content. Such an architecture is preferably based on the usage of scalable bitstreams and a format-agnostic content adaptation engine. To obtain fully embedded scalable bitstreams, the Joint Scalable Video Model (JSVM) has been used in this paper. Hereby, JSVM defines a scalable extension on top of the H.264/AVC specification. This extension will make it possible to create bitstreams that are scalable along the temporal, spatial, and SNR axis. On the other hand, bitstream structure descriptions can be used to realize an elegant and format-agnostic adaptation engine. Such descriptions can be created by making use of the MPEG-21 Bitstream Syntax Description Language (BSDL) standard. The latter allows to describe the high-level structure of scalable bitstreams in XML. This paper explains how fully scalable bitstreams can be customized by transforming BSDL-based bitstream structure descriptions. From our performance analysis, one can conclude that the transformation of the XML description, as well as the generation of the adapted bitstream, can be done several times faster than real time.**

I. Introduction

Nowadays, our pervasive multimedia ecosystem allows that multimedia content can be accessed by different users from a various collection of terminals and networks. For instance, thanks to the growing processing power, Personal Digital Assistants (PDAs) and cellular phones are already able to decode high quality video sequences. In such a diverse environment, it is necessary to control the huge miscellany of content and resource constraints such as terminal capabilities, band width, CPU power, etcetera. Therefore, two important technologies are indispensable to obtain such a multimedia environment, in particular scalable bitstreams and a standardized format-agnostic content adaptation framework. These two technologies belong to different research topics: scalable bitstreams are a part of media coding techniques while the functioning of an adaptation framework rather belongs to the metadata community. This paper describes how these two different worlds can be brought together in order to shift the focus of the content customization process to the high-level XML domain, hereby taking into account the different usage environment parameters (e.g., the CPU power of a terminal). Because of the fact that motion pictures are playing an increasingly important role in our multimedia environments, the focus of this paper will be put on video sequences as multimedia resources.
To obtain fully embedded scalable bitstreams, the Joint Scalable Video Model (JSVM) specification is used. JSVM is based on the successful H.264/AVC standard and it describes an extension mechanism such that scalable bitstreams can be obtained. This is in contrast with the original H.264/AVC specification that is not designed to produce scalable bitstreams. On the other hand, the MPEG-21 Bitstream Syntax Description Language (BSDL) is used to describe the structure of a bitstream in XML such that the adaptations can be expressed in the XML domain instead of in the low-level compressed domain. This paper describes how XML descriptions can be generated for JSVM encoded bitstreams and how the embedded scalability can be exploited in this high-level domain.
The outline of the paper is as follows. In Section II, the global structure of JSVM will be described as well as how the different embedded scalability axes can be found in the generated bitstreams. Section III discusses the construction of a BS Schema for a JSVM encoded bitstream. The implementation of the different stylesheets to obtain descriptions of partial streams with a decreasing quality along one or more scalability axes is described in Section IV. A performance analysis of the format-agnostic framework used is provided in Section V. Finally, a conclusion is given in Section VI.

II. Joint Scalable Video Model

The Joint Video Team (JVT) has started the standardization of a new scalable video specification in 2004 [1]. Scalable video coding schemes are able to encode the input sequence once at the highest resolution, frame rate, and visual quality, after which it is possible to extract partial streams containing a lower quality. The bitstream extractor has to generate the partial streams in an efficient way, which means that no decode-encode steps of the chroma and luma data of the original bitstream are needed. Every scalability axis has to be independently accessible. The three scalability axes are temporal, spatial, and SNR (Signal-to-Noise Ratio). A reduction of the quality along the temporal axis results in a decreasing frame rate; along the spatial axis in a smaller spatial resolution; and along the SNR axis in a lower visual quality.
The new standard under development, in particular JSVM, is an extension of the single-layered H.264/MPEG-4 Advanced

Video Coding scheme (H.264/AVC). This results in the requirement that the base layer of the scalable bitstream should be H.264/AVC compliant [2]. The structure of a possible encoder, providing three spatial levels, is given in Fig. 1. In

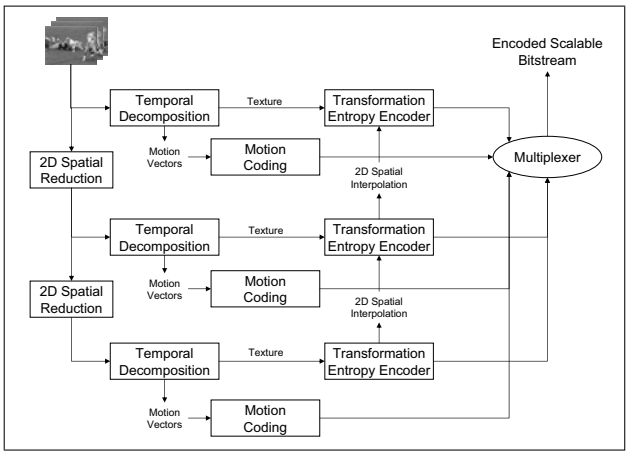

Fig. 1. JSVM encoder structure for providing three spatial levels, [5].

this figure, one can see that the original input video sequence has to be downscaled in order to obtain the different spatial layers (resulting in spatial scalability). For each spatial layer a temporal decomposition is performed leading to temporal scalability. The temporal scalability can be achieved in two ways, in particular by using hierarchical B pictures or by using Motion Compensated Temporal Filtering (MCTF). Both temporal decompositions lead to a motion field and texture data. The layered structure of the JSVM contains the possibility to use motion information and texture encoding of lower spatial layers for predicting the information in the higher layers. Finally, the texture information is spatially transformed and entropy encoded by using Fine or Coarse Grain Scalability (FGS and CGS) to obtain the SNR scalability axis.

The structure of a bitstream generated by a coding scheme as given in Fig. 1 is depicted in Fig. 2. Every scalable bitstream starts with a Supplemental Enhancement Information (SEI) message. Such a message contains information about the scalability axes incorporated in the bitstream such as the number of spatial levels, the temporal decomposition, the spatial resolution of the base layer, the frame rate, etc. An SEI message can be ignored by a decoder to reproduce the luma and chroma samples and is only necessary to assist the extractor in generating partial bitstreams [8]. These SEI messages are very important to satisfy the requirement to support efficient bitstream extraction. After the SEI message, a number of Sequence Parameter Sets (SPS) follow; in particular at least one SPS is needed for every spatial layer. An SPS is applicable to a complete sequence of pictures of a particular spatial layer and contains information about the profile used, the spatial resolution of the pictures in the sequence, etc. A number of Picture Parameter Sets (PPS) are also encapsulated in the bitstream. A PPS applies to a number of pictures of a sequence and contains information such as the type of the entropy encoding used, the presence of a deblocking filter, etc. Because of the fact that every PPS must refer to an SPS, there are at least as many PPSs as SPSs and mostly there are more PPSs than SPSs. Finally, the NALUs (Network Abstraction Layer Units), containing the luma and chroma information, are integrated into the bitstream. Every unit starts with a header followed by the actual payload, which is nothing more than a concatenation of entropy encoded MacroBlocks (MBs). The NALU header contains the type of the unit. When the slice data of the unit belongs to an extension layer, scalability information can be present in the header as well.

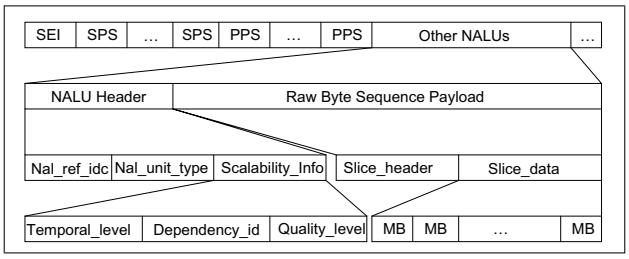

Fig. 2. Structure of a scalable bitstream.

III. MPEG-21 BSDL for JSVM bitstreams

BSDL is embedded in part 7 of the MPEG-21 specification; this part is better known as Digital Item Adaptation (DIA). MPEG-21 describes a multimedia framework which aims to enable the transparent and augmented use of multimedia resources across a wide range of networks and devices [6]. The DIA standard specifies tools that describe terminal characteristics, network capabilities, and user preferences. The specification also contains two tools for describing the high-level structure of an encoded bitstream, in particular BSDL and generic Bitstream Syntax Schema (gBS Schema). In this paper, we use MPEG-21 BSDL as a language for describing the high-level structure of compressed bitstreams.

MPEG-21 BSDL is based on W3C XML Schema and is developed to automatically generate Bitstream Syntax Descriptions (BSDs). A BSD describes the high-level structure of a bitstream in XML. This XML-based description can then be transformed in order to reflect a desired adaptation of a scalable bitstream, and can subsequently be used to create a customized version of the bitstream. The general functioning of the BSDL architecture is given in Fig. 3. Dependent on the coding specification used, a BS Schema can be developed such that it describes the high-level structure of a compressed bitstream. In Fig. 3, one can see that a BSD can be generated by a format-independent parser once the original (encoded) bitstream and corresponding BS Schema is known. The functioning of the BintoBSD Parser is described in the standard because of the fact that a BS Schema can only contain elements that are fixed by BSDL. Once an XML-based BSD is generated, the description can be transformed by using a ubiquitous transformation technology such as XSLT (Extensible Stylesheet Language: Transformations) [3] or STX (Streaming Transformations for XML) [4]. The result of the

transformation is an adapted XML description that is still valid against the BS Schema of the coding scheme used. From this description, it is possible to generate an adapted bitstream by using the transformed BSD; the original bitstream; and the corresponding BS Schema. The generic BSDtoBin Parser can be used to realize this process and the functioning of this tool is also described in the MPEG-21 DIA specification. This tool is a format-agnostic bitsteam generation tool, i.e. the code base of this parser does not have to be rewritten in order to support the customization of another coding format.

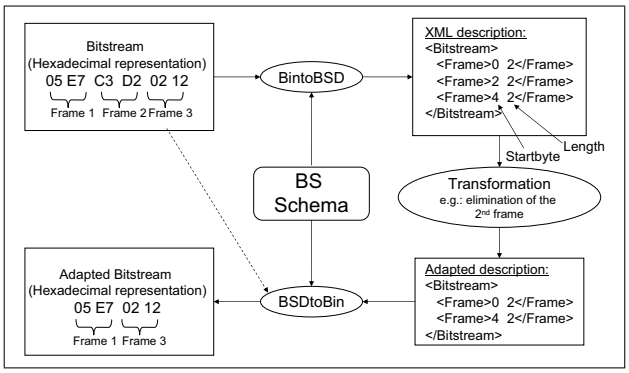

Fig. 3. Functioning of the BSDL framework.

In this paper, we use bitstreams that are compliant with the third version of JSVM [5]. The structure of those bitstreams is explained in Section II. The following syntactical datastructures are described in XML: SEI, SPS, PPS, and the NALU header. One can see that we do not describe the encoded residual data but only the necessary header information. The actual payload can be obtained by pointing to blocks of data in the original bitstream (resulting in a high-level description). Therefore, one has to use the `byteRange` datatype that is part of the BSDL specification and that is not present in the W3C XML Schema standard. Other datatypes that were added to BSDL are `fillByte` and bit-based datatypes. Moreover, BSDL offers the possibility to design new datatypes by deriving them from the W3C XML Schema datatypes or from the BSDL built-in datatypes. Nevertheless, some syntax elements are having a datatype that cannot be described by BSDL such as the exponential Golomb datatype. To parse a syntax element that is represented by this datatype, we have used a non-normative extension of the BSDL standard, in particular the `implementation` attribute in the BS Schema. This attribute allows to rely on procedural objects in order to perform complex computations or to deal with complex datatypes by calling Java classes from the BS Schema [6].

IV. Exploiting the scalability in XML

An adaptation of the original scalable bitstream along the three scalability axes has to be realized by transforming the generated BSD. We have implemented the XML transformations by using XSLT. In Fig. 4, a part of a generated BSD is represented. An example of the four most important parts of the BSD is given, in particular the SEI message, an

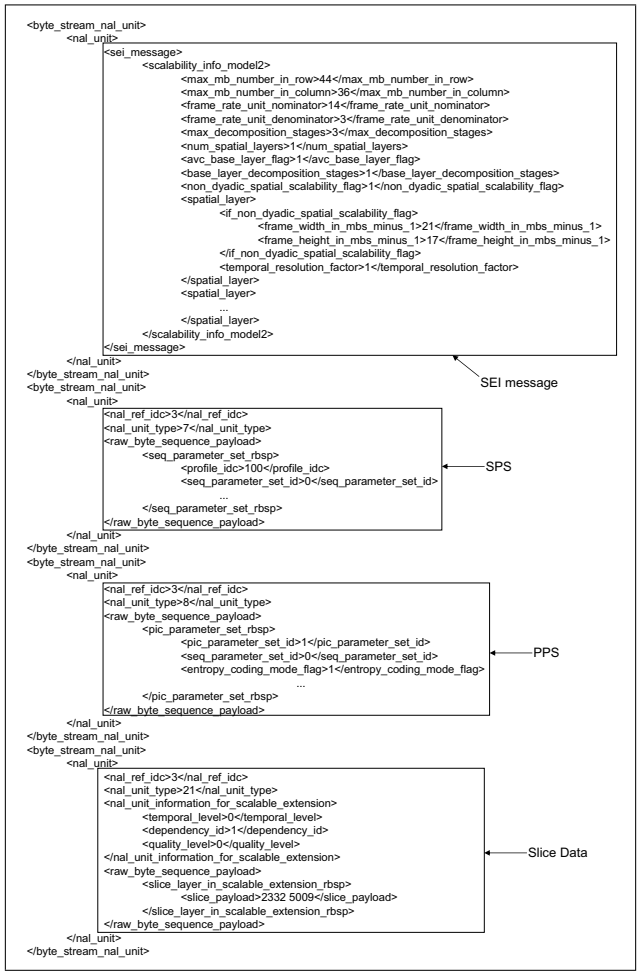

Fig. 4. Example of a BSD containing the necessary information to realize the different scalability transformations.

SPS, a PPS, and an example of a NALU containing slice data that belongs to a frame that is part of an enhancement layer. From the SEI message, one can detect that the corresponding bitstream contains 2 spatial layers (a base layer and an enhancement layer, indicated by 1), 4 temporal levels (indicated by the 3 enhancement temporal levels), and that the highest spatial layer contains pictures of 44 MBs by 36 MBs (resulting in a frame resolution of 704x576 pixels). Based on the `profile_idc` syntax element of an SPS, the extractor can decide to which spatial layer the SPS belongs. The same is applied for the PPS but the reference to the SPS gives the indication to which spatial layer the PPS belongs. Finally, the last NALU contains the actual encoded slice data. In this example, one can see that the NALU contains scalability information such as the temporal, spatial, and quality level. It is clear that the slice in this unit belongs to the second spatial layer and to the base temporal and the base quality layer. In case that this information is not present in the NALU, the slice data belongs to the base spatial layer. One can also see how the `byteRange` datatype is used to point to a block of data in the original bitstream, in particular to a block that starts at

TABLE I
PERFORMANCE MEASUREMENTS.

Bitstream				BintoBSD (s)		Size	Transformations (s)			BSDtoBin (s)
#Frames	#Temp Layers	#Spat Layers	Size (Kb)	Ref Software	Modified Version	BSD (Kb)	Temp	Spat	SNR	
60	4	2	558	39.4	15.3	172	0.28	0.30	0.33	0.13
64	4	3	485	112.9	41.3	302	0.43	0.42	0.41	0.20
150	5	2	2058	130.4	49.3	322	0.34	0.44	0.45	0.21
300	5	2	1489	417.3	166.9	631	0.42	0.54	0.57	0.22

byte position 2332 and that has a length of 5009 bytes.
Based on the discussed information, it is clear that it is possible to implement a stylesheet that removes the necessary NALUs in order to obtain a description that is linked to a bitstream that has a lower spatial, temporal, or SNR quality.

V. EXPERIMENTAL RESULTS

In this section, we discuss the performance evaluation of the BSDL-based adaptation framework in the context of scalable bitstreams that are compliant with JSVM. We have measured the processing time of the BSDL Parsers, in particular of the BintoBSD and BSDtoBin Parser, as well as the time needed to execute the transformations in the XML domain. We have used two implementations of the BSDL software to obtain the execution times. The first implementation is version 1.2.1 of the MPEG-21 BSDL reference software, while the second implementation contains some own-developed optimizations. Most of the execution time of the BintoBSD Parsers is spent on the evaluation of XPath expressions. In our modified version, the process to evaluate the XPath expressions is implemented in a faster manner, in particular by using the Xalan library as efficient as possible. The transformations used remove two temporal levels, one spatial layer, or eliminate the enhancement quality layer.
The measurements were done on a PC having an Intel Pentium IV CPU, clocked at 2.8GHz with Hyper-Threading and having 1GB of RAM at its disposal. The bitstreams were created by relying on the JSVM reference software version 3.
The results of our performance analysis are given in Table I. The first columns contain the characteristics of the scalable bitstreams used, such as the size, the number of frames, spatial layers, and temporal levels. The execution time of the BintoBSD Parser to obtain the BSDs is high, certainly in comparison with the length of the bitstream and the generation of the bitstream by the BSDtoBin tool. One can see that our modified version of the reference software needs less time to generate the same BSD. Using the algorithm as discussed in [7] should lead to lower execution times, but this is not the main topic of this paper. Further, the XSLT transformations can be executed very fast. The kind of scalability that is exploited does not have an impact on the execution times.

VI. CONCLUSIONS

In this paper, we have described a harmonized approach between the use of scalable video coding and a metadata-driven content adaptation engine. Therefore, in order to obtain fully embedded scalable bitstreams, the Joint Scalable Video Model (JSVM) was used. The latter is an emerging video specification that is based on H.264/AVC. To extract partial bitstreams, containing a lower spatial, temporal, or SNR quality, we have described the high-level structure of the generated bitstream in XML by using the MPEG-21 Bitstream Syntax Description Language standard. For the first time, the adaptation along a scalability axis of the original encoded bitstream is realized by transforming the corresponding XML description. During a performance analysis of the framework, one can conclude that the generation of the description takes a lot of time. However, once the description is generated, a process that only has to be executed once, the transformation of the description as well as the subsequent generation of the adapted bitstream from the transformed description can be executed very fast. Finally, we can conclude that JSVM is a good candidate to be described in XML and that the generated descriptions can be used in a standardized multimedia format-agnostic content adaptation framework such as MPEG-21.

ACKNOWLEDGMENT

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), the Belgian Federal Science Policy Office (BFSPO), and the European Union.

REFERENCES

[1] Applications and Requirements for Scalable Video Coding, ISO/IEC JTC1/SC29/WG11/N6880, January 2005.
[2] ITU-T and ISO/IEC JTC 1, Advanced video coding for generic audiovisual services, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC (2003).
[3] M. Kay, *XSLT Programmer's Reference, 2nd Edition*. Wrox Press Ltd. Birmingham, UK, 2001.
[4] P. Cimprich, *Streaming Transformations for XML (STX) Version 1.0 Working Draft*, http://stx.sourceforge.net/documents/spec-stx-20040701.html, July, 2004.
[5] J. Reichel, M. Wien, H Schwarz, *Joint Scalable Video Model JSVM-3*, Doc. JVT-P202, July 2005.
[6] D. De Schrijver, C. Poppe, S. Lerouge, W. De Neve, R. Van de Walle, *MPEG-21 Bitstream Syntax Descriptions for Scalable Video Codecs*, Accepted for publication in Multimedia Systems Journal.
[7] D. De Schrijver, W. De Neve, K. De Wolf, and R. Van de Walle, *Generating MPEG-21 BSDL Descriptions Using Context-Related Attributes*, Proceedings of the 7th IEEE International Symposium on Multimedia. Irvine (CA, USA) Dec 2005, pp. 79 - 86.
[8] W. De Neve, D. Van Deursen, D. De Schrijver, K. De Wolf, and R. Van de Walle, *Using Bitstream Structure Descriptions for the Exploitation of Multi-layered Temporal Scalability in H.264/AVC's Base Specification*, Proceedings of the 2005 Pacific - Rim Conference on Multimedia, Springer-Verlag, Jeju, Korea, 2005, pp. 641 – 652.