XML-driven Exploitation of Combined Scalability in Scalable H.264/AVC Bitstreams

Davy De Schrijver, Wesley De Neve, Koen De Wolf, Peter Lambert, Davy Van Deursen, and Rik Van de Walle
Department of Electronics and Information Systems – Multimedia Lab
Ghent University – IBBT
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium
Email: davy.deschrijver@ugent.be

Abstract— **The heterogeneity in the contemporary multimedia environments requires a format-agnostic adaptation framework for the consumption of digital video content. Scalable bitstreams can be used in order to satisfy as many circumstances as possible. In this paper, the scalable extension on the H.264/AVC specification is used to obtain the parent bitstreams. The adaptation along the combined scalability axis of the bitstreams is done in a format-independent manner. Therefore, an abstraction layer of the bitstream is needed. In this paper, XML descriptions are used representing the high-level structure of the bitstreams by relying on the MPEG-21 Bitstream Syntax Description Language standard. The exploitation of the combined scalability is executed in the XML domain by implementing the adaptation process in a Streaming Transformation for XML (STX) stylesheet. The algorithm used in the transformation of the XML description is discussed in detail in this paper. From the performance measurements, one can conclude that the STX transformation in the XML domain and the generation of the corresponding adapted bitstream can be realized in real time.**

I. Introduction

Nowadays, digital video content can be accessed by different users in heterogeneous environments. Two components, in particular scalable bitstreams and a format-agnostic adaptation framework, are needed in order to control the huge diversity in content and resource constraints such as terminal capabilities, band width, and user preferences. In this paper, both technologies are brought together to adapt the scalable bitstreams by making use of a format-agnostic engine.

The aim of Scalable Video Coding (SVC) is to encode a video sequence once, after which the generated bitstream can be adapted by using simple truncation operations. These operations make it possible to extract bitstreams containing a lower frame rate, spatial resolution, and/or visual quality from the original coded bitstream. Therefore, an SVC bitstream contains three scalability axes (temporal, spatial, and SNR) along which adaptations can be executed. Every scalability axis is independently accessible but it is also possible to adapt the bitstream by truncating along multiple axes at the same time. This results in *combined scalability* and this type of scalability will be exploited in this paper. Hereby, we make use of bitstreams compliant with the Joint Scalable Video Model (JSVM) version 6 specification [1].

The scalable bitstreams will be adapted by a format-independent engine. Therefore, we will describe the high-level structure of the bitstreams in XML. These XML descriptions will be generated by relying on the MPEG-21 Bitstream Syntax Description Language (MPEG-21 BSDL, [2]) framework. This gives us the possibility to shift the focus of the content customization process to the XML domain. The adaptation process in the XML domain can be realized by a transformation engine without knowledge of the underlying coding format. Such an engine typically takes a stylesheet, representing the transformation actions, as input. In this paper, we will make use of Streaming Transformations for XML (STX, [3]) and we will pay special attention to the implementation of an STX stylesheet exploiting the combined scalability characteristic of JSVM6-coded bitstreams.

The performance of doing the adaptation process in the XML domain will be investigated in order to verify the usefulness of a fully XML-driven adaptation framework for scalable video content adaptation.

The outline of this paper is as follows. In Section II, a fully XML-driven adaptation framework is explained. The high-level structure of the scalable bitstreams is discussed in Section III. Section IV describes the adaptation process in the XML domain. More precisely, the STX stylesheet implementing the combined scalability is discussed. The performance results and accompanying discussing are provided in Section V. Finally, a conclusion is given in Section VI.

II. XML-driven adaptation framework

The MPEG-21 Digital Item Adaptation (DIA) standard enables the customization of multimedia content in heterogeneous environments. One of the building blocks of DIA is MPEG-21 BSDL. This language allows to build an interoperable description-driven framework in which multimedia content can be adapted in a format-agnostic manner [4]. In Fig. 1, an overview of our fully XML-driven framework for video content adaptation is given. The core of the framework is the automatic generation of XML descriptions containing information about the high-level structure of the scalable bitstreams and the adaptation thereof. Explanatory notes of the indicated numbers in Fig. 1 are provided below.

(1) The Bitstream Syntax Schema (BS Schema) represents the high-level structure of the coding format of the scalable bitstream as specified in an international standard. The language used to construct such a BS Schema is BSDL, which is standardized in the DIA specification.

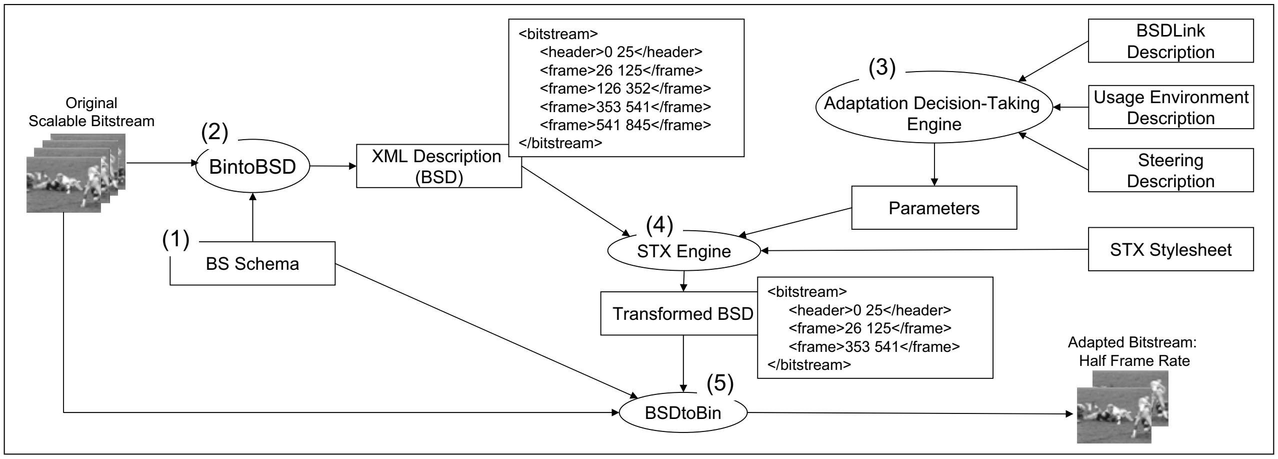

Fig. 1. Overview of a fully XML-driven framework for video content adaptation based on MPEG-21 DIA

(2) The Bitstream Syntax Description (BSD) of a (scalable) bitstream is an XML document and generated by a format-agnostic parser. This parser, i.e. the BintoBSD Parser, takes the original bitstream and accompanying BS Schema as input to generate the BSD.

(3) A scalable bitstream has to be adapted according to a usage environment. Therefore, an engine is needed to provide an adequate decision of the properties of the scalability axes along which the adaptation has to be executed. This Adaptation Decision-Taking Engine (ADTE) makes use of the Usage Environment Description (UED) describing the terminal capabilities, network characteristics, and user preferences. Because of the format-agnostic nature of this engine, the ADTE has to know which adapted versions can be generated from the original bitstream. This information about the scalability possibilities of the bitstream is described in the steering description. The outcome of the ADTE is a set of parameters, which have to be transmitted to the XML transformation engine. The BSDLink description defines these parameters and gives the ADTE the necessary information to map the correct values to the parameters.

(4) Once the ADTE has stipulated the adaptation parameters, the generated BSD can be transformed by using a ubiquitous transformation language like STX. The adaptation is represented by a stylesheet and the format-neutral STX engine uses the parameters to generate the transformed BSD. This means that the adaptation is executed in the XML domain instead of on the bitstream itself.

(5) The format-agnostic BSDtoBin Parser creates an adapted bitstream, using the transformed BSD, the corresponding BS Schema, and the original (scalable) bitstream.

Note that the 4 engines in the framework are format-agnostic meaning that the code base of these parsers does not have to be rewritten to support the adaptation of bitstreams compliant with other coding formats. Furthermore, all communication (i.e., the descriptions) within the framework is XML driven. It is the first time that a fully XML-driven framework is evaluated in which JSVM-coded bitstreams are used.

III. Scalable Extension of H.264/MPEG-4 AVC

The video coding specification used, in particular JSVM6, is an extension of the non-scalable single-layered H.264/MPEG-4 Advanced Video Coding scheme (H.264/AVC). Consequently, a JSVM decoder can decode H.264/AVC bitstreams and the base layer of a scalable bitstream should be compliant with H.264/AVC. Note, the fundamental building blocks of JSVM bitstreams are Network Abstraction Layer Units (NALUs), similar to H.264/AVC bitstreams.

Fig. 2 shows the structure of a JSVM6-coded bitstream providing three temporal levels, two spatial layers, and a quality enhancement layer. This means that a JSVM6-encoder generates scalable bitstreams containing following scalability properties along which adaptations can be executed at the same time (i.e., combined scalability).

- Temporal scalability is obtained by using hierarchical B picture.
- By down-sampling the original video sequence, spatial scalability is obtained.
- To obtain quality or SNR scalability, one has to additionally code quality enhancement layers on top of a quality base layer. The JSVM6 specification allows Coarse and Fine Grain Scalability (CGS and FGS). In case of CGS, a complete enhancement layer is removed. While in case of FGS, the enhancement layers can be truncated at any arbitrary byte position.

In in-depth explanation of the structure of JSVM6-coded bitstreams is discussed in [1].

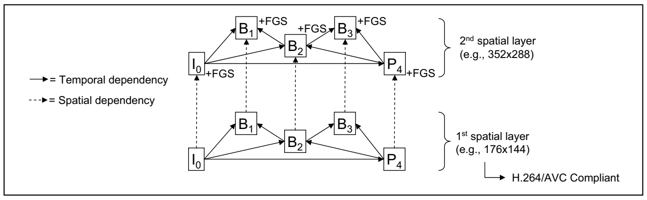

Fig. 2. Structure of a JSVM-coded bitstream

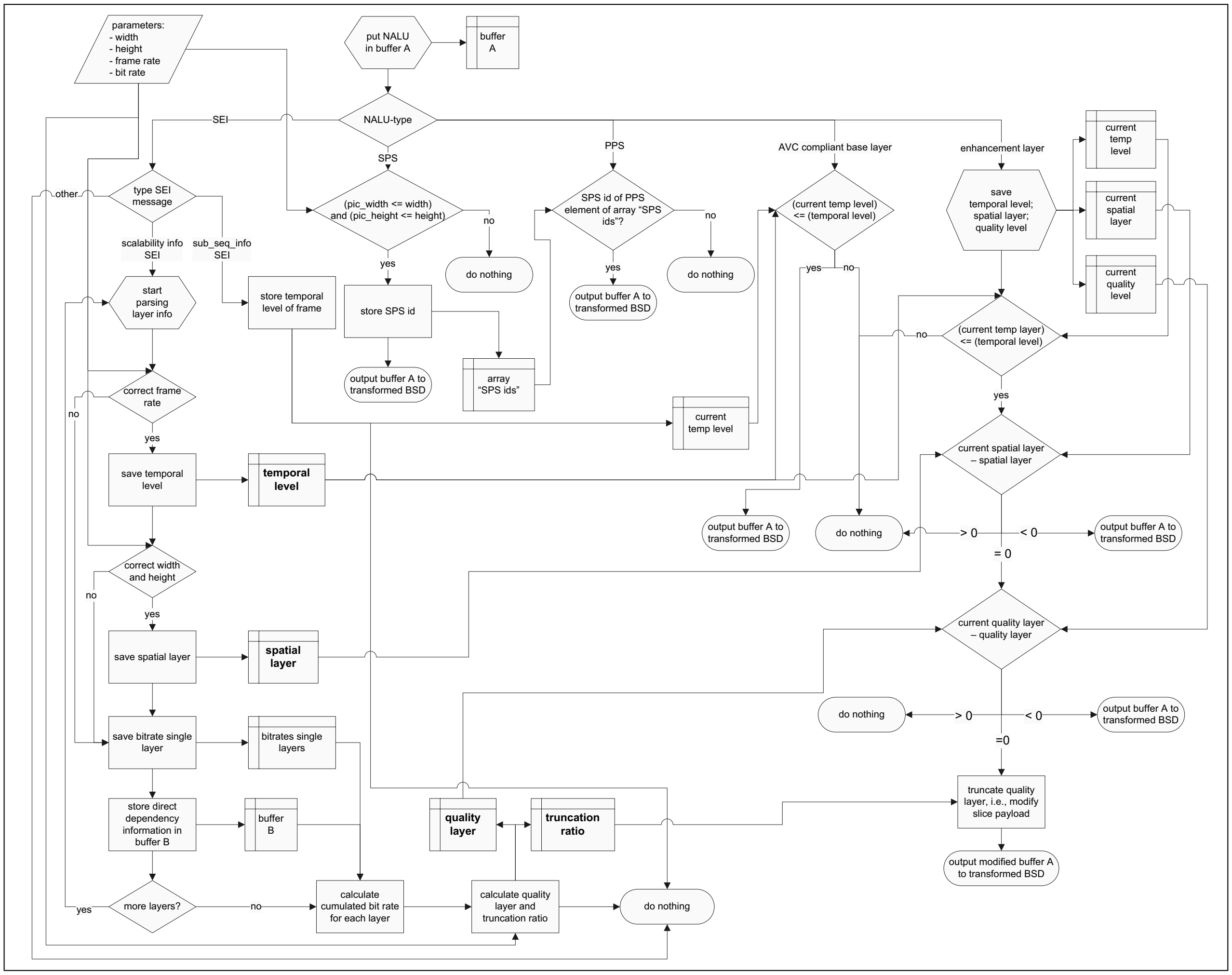
parameters:
- width
- height
- frame rate
- bit rate
put NALU in buffer A
buffer A
NALU-type
SEI
SPS
PPS
AVC compliant base layer
enhancement layer
other
type SEI message
scalability info SEI
sub_seq_info SEI
(pic_width <= width) and (pic_height <= height)
no
yes
do nothing
store SPS id
output buffer A to transformed BSD
array "SPS ids"
SPS id of PPS element of array "SPS ids"?
yes
no
output buffer A to transformed BSD
do nothing
(current temp level) <= (temporal level)
yes
no
save temporal level; spatial layer; quality level
current temp level
current spatial layer
current quality level
(current temp layer) <= (temporal level)
no
yes
current spatial layer – spatial layer
output buffer A to transformed BSD
do nothing
> 0
< 0
= 0
output buffer A to transformed BSD
current quality layer – quality layer
do nothing
> 0
< 0
=0
output buffer A to transformed BSD
truncate quality layer, i.e., modify slice payload
output modified buffer A to transformed BSD
start parsing layer info
store temporal level of frame
current temp level
correct frame rate
no
yes
save temporal level
temporal level
correct width and height
no
yes
save spatial layer
spatial layer
save bitrate single layer
bitrates single layers
store direct dependency information in buffer B
buffer B
quality layer
truncation ratio
more layers?
yes
no
calculate cumulated bit rate for each layer
calculate quality layer and truncation ratio
do nothing

Fig. 3. Data flow of our STX stylesheet for exploiting combined scalability

IV. Exploitation of Combined Scalability in XML

The combined scalability of a JSVM bitstream can be exploited when a desired frame rate, spatial resolution, and bit rate are given to the STX transformation engine, which are the parameters in Fig. 1. This engine is steered by an STX stylesheet representing the adaptation in the XML domain and transforms the generated BSD. To obtain such a BSD, a BS Schema for the JSVM6 specification is developed which we discuss in detail in [5]. This BS Schema describes the necessary information in XML in order to execute the combined scalability. More precisely, the following syntactical data structures are described: Supplemental Enhancement Information (SEI) messages, Sequence Parameter Sets (SPSs), Picture Parameter Sets (PPSs), and the NALU headers.

The data flow of our STX stylesheet for the exploitation of the combined scalability is shown in Fig. 3. As one can observe, the adaptation process can be divided into three main parts.

In the first step, the scalability information SEI message is interpreted. Based on the information encapsulated in this SEI message, the desired temporal, spatial, and quality level is determined. Furthermore, the truncation ratio is calculated in case the last quality layer is an FGS layer. The obtained values are stored in the stylesheet and are used during the decision step of the NALUs representing slices of the bitstream.

In the second step, the parameter sets are parsed. Only the SPSs needed to decode the desired spatial layer will be kept in the BSD and the other SPSs are removed from the description. Because each PPS refers to an SPS (using the `sequence_parameter_set_id` syntax element), we can also remove all PPSs referring to removed SPSs.

In the third step, the NALUs representing slices of the bitstream are parsed. Hereby, we have to make a distinction between slices belonging to an AVC-compliant base layer and belonging to an enhancement layer. The main difference is that the NALUs of the AVC-compliant base layer contain no scalability information. Therefore, a `sub_seq_info` SEI message precedes these NALUs containing information of the temporal level. The NALUs of the enhancement layers contain the information about the temporal, spatial, and quality level in the NALU header. Based on these values and the internal stored values from in the first step, a decision can be made to remove, to keep, or to truncate the NALU in question.

TABLE I

PERFORMANCE RESULTS OF THE ADAPTATION ENGINE

Original Bitstream			BintoBSD Parser			STX Transformation				BSDtoBin Parser	
Name	#Frames	Size (KB)	ET (s)	size_p (KB)	size_c (KB)	Desired Bitstream	ET (s)	size_p (KB)	size_c (KB)	ET (s)	OB (Kbit/s)
Seq_1	75	3389	4.35	601	12.3	176x144@15:120	0.77	36.5	3	0.76	130.38
						352x288@30:700	0.83	133	4	0.87	740.56
						704x576@60:5000	0.92	392	8	0.99	5334.63
Seq_2	150	5980	6.94	1072	20.7	176x144@15:120	1.04	67	3	0.80	127.79
						352x288@30:700	1.14	325	7	0.95	727.45
						704x576@60:5000	1.31	764	13	1.10	5269.10
Seq_3	300	11451	12.23	2013	37.2	176x144@15:120	1.43	126	4	0.81	121.24
						352x288@30:700	1.75	637	11	1.04	707.79
						704x576@60:5000	2.10	1509	24	1.33	5142.94
Seq_4	480	19966	19.44	3148	57.3	176x144@15:120	2.12	198	5	0.99	119.81
						352x288@30:700	2.36	781	13	1.10	700.42
						704x576@60:5000	2.98	2404	37	1.58	5125.12

V. PERFORMANCE RESULTS

Methodology. To evaluate the performance of our XML-driven adaptation framework, we have generated four JSVM6-compliant scalable bitstreams. Each bitstream contains a part of the well-known *ice* sequence with a resolution of 704×576 at a frame rate of 60Hz. The encoder generates bitstreams with 5 temporal, 3 spatial, and 4 quality levels.

For each bitstream, the corresponding BSD is generated by using an optimized BintoBSD Parser as explained in [6]. These BSDs are subject to the transformation reflecting the adaptation in the XML domain. From each bitstream, 3 partial streams are generated containing a resolution of 176×144 at 15Hz (120 Kbit/s), a resolution of 352×288 at 30Hz (700 Kbit/s), and a resolution of 704×576 at 60Hz (5000 Kbit/s). The STX engine used is *Joost* (version 2005-05-21). Finally, a modified BSDtoBin Parser of the MPEG-21 reference software version 1.2.1 is used to generate the adapted bitstreams.

Discussion of the results. The performance results of the different steps are given in Table I. The BintoBSD Parser is the first engine of which the performance was investigated. The Execution Times (ETs) of the parser, the sizes of the resulting XML descriptions in plain text (size_p), and the sizes of the compressed BSDs (size_c by using EasyZip v3.5) are given in the table. Hereby, we can conclude that the ET is linear as a function of the length of the sequence. The sizes of the plain-text generated BSDs is substantially compared to the original bitstream, approximately 15% of the size of the bitstream. By compressing the BSDs, the overhead originates from the XML description is negligible, roughly 0.3%.

The STX engine is the next step in the adaptation process that was measured. The ETs are linear as a function of the length of bitstream if the sequence is long enough (meaning that the start-up time can be ignored). The sizes (in plain text and compressed) of the transformed BSDs represent the influence of the adaptation parameters on the available NALUs.

Finally, the ETs of the BSDtoBin Parser are given resulting in a fast engine. Note, the transformation together with the generation of the adapted bitstream is realized in almost real time. The Obtained Bit rates (OBs) of the adapted bitstreams approach the desired rates very well. This means that our adaptation engine can generate bitstreams containing a desired bit rate without knowledge of the underlying coding format.

VI. CONCLUSIONS

In this paper, a format-agnostic framework for scalable video content adaptation was proposed in which all communication is based on XML descriptions. Not only the usage environment but also the high-level structure of the scalable bitstreams is described in XML by using MPEG-21 BSDL. This gives us the opportunity to shift the adaptation process to the XML domain. The JSVM6-coded bitstreams can be adapted along the three scalability axes at the same time (i.e., combined scalability). The transformation exploiting the combined scalability of the XML descriptions was implemented in STX. From the performance results, we can conclude that the execution time of the transformation and the generation of the adapted bitstreams is linear as a function of the length of the sequences. Furthermore, we have proven that our XML-driven framework can execute the adaptations in real time.

ACKNOWLEDGEMENTS

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] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model JSVM-6," *Doc. JVT-S202*, April 2005.

[2] G. Panis, A. Hutter, J. Heuer, H. Hellwagner, H. Kosch, C. Timmerer, S. Devillers, and M. Amielh, "Bitstream syntax description: a tool for multimedia resource adaptation within MPEG-21," *Signal Processing: Image Communication*, vol. 18, no. 8, pp. 721–747, 9 2003.

[3] O. Becker, "Transforming XML on the fly," in *Proceedings of XML Europe*, May 2003.

[4] S. Devillers, C. Timmerer, J. Heuer, and H. Hellwagner, "Bitstream syntax description-based adaptation in streaming and constrained environments," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 463–470, 6 2005.

[5] D. De Schrijver, W. De Neve, K. De Wolf, S. Notebaert, and R. Van de Walle, "XML-based customization along the scalability axes of H.264/AVC scalable video coding," in *Proceedings of IEEE ISCAS 2006*, Island of Kos, Greece, 5 2006, pp. 465–468.

[6] D. De Schrijver, W. De Neve, K. De Wolf, and R. Van de Walle, "Generating MPEG-21 BSDL descriptions using context-related attributes," in *Proceedings of the 7th IEEE ISM*, Irvine, (CA, USA), 12 2005, pp. 79–86.