XML-based Exploitation of Region of Interest Scalability in Scalable Video Coding

Davy De Schrijver, Wesley De Neve, Davy Van Deursen, Yves Dhondt, 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 use of Regions Of Interest (ROIs) is a useful concept for many application scenarios, especially for those applications that are deployed in heterogeneous multimedia environments. In this paper, we show how Flexible Macroblock Ordering can be used in the scalable extension of the H.264/AVC specification in order to define the ROIs in the coded bitstream. Furthermore, we introduce an XML-driven adaptation framework based on the MPEG-21 Bitstream Syntax Description Language in order to implement the ROI extraction process. This framework gives us the opportunity to adapt scalable bitstreams by using an engine that has no knowledge of the underlying coding format. From the performance analysis of our adaptation framework, we can conclude that the ROIs can be extracted in the XML domain and that the ROIs in the adapted bitstream are still intact without quality degradation. Furthermore, the traditional drifting problem caused by the ROI extraction can be neglected. Finally, we show that the adaptation process in the XML domain can be executed in real time.**

I. Introduction

Nowadays, our pervasive multimedia systems allow that video content can be accessed by different users from a various collection of terminals and networks. Furthermore, the more restricted the devices are, the more important the Regions Of Interest (ROIs) are in the video sequences. Consequently, this paper will investigate how ROI scalability can be obtained in digital video. Therefore, two important technologies are indispensable to obtain such an adaptive multimedia environment, in particular scalable video bitstreams and a standardized format-agnostic content adaptation framework.

To obtain the ROI scalability, we have used the Scalable extension of the H.264/AVC (SVC) specification [1]. In order to define the ROIs during the encoding phase, the Flexible Macroblock Ordering (FMO) tool has been used. In fact, FMO is intended to be used as an error resilience tool, but we show that it also can be used to extract ROIs from a bitstream. This approach is outlined in Section 2.

As previously mentioned, we want to do the adaptation in a format-agnostic manner. Therefore, it is preferable to realize the extraction process of the ROIs in the XML domain instead of immediately on the bitstream itself [2]. The MPEG-21 Bitstream Syntax Description Language (BSDL) framework can be used to realize the content adaptation process in the XML domain. In this paper, we describe how MPEG-21 BSDL can be used to create XML descriptions of the high-level structure of the scalable bitstreams and how the adapted bitstreams, in which the ROIs are extracted, can subsequently be generated using transformed XML documents. This adaptation process in the XML domain is explained in Section 3.

Furthermore, the complete framework is evaluated in terms of execution times and visual quality of the adapted bitstreams in Section 4. Finally, Section 5 concludes this paper.

II. ROI Scalability with FMO in SVC

SVC is an extension of the single-layered H.264/AVC specification, meaning that all tools of H.264/AVC are also available in its scalable extension. One of these tools is the error resilience tool *Flexible Macroblock Ordering (FMO)*. FMO allows to code the macroblocks of a picture in an order other than raster scan. Therefore, a number of independent *slice groups* can be defined and each macroblock has to be mapped on exactly one slice group. This mapping is described by a MacroBlock Allocation map (MBAmap), which is conveyed by a Picture Parameter Set (PPS) in the bitstream. In order to avoid that the complete MBAmap has to be encapsulated in the PPS, which introduce a considerable amount of overhead, the H.264/AVC standard has described 6 predefined FMO types.

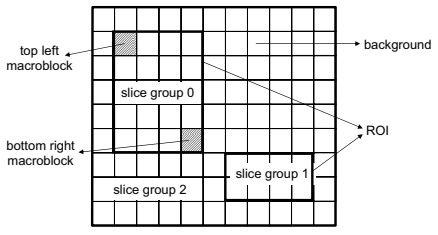

Fig. 1. ROI coding by using FMO type 2

By using FMO, it is possible to define a ROI during the encoding phase. Therefore, we have used FMO type 2. This type defines one or more rectangular slice groups and a background. The rectangular slice groups represent the ROIs. Fig. 1 represents a frame containing two ROIs where each ROI is represented by a slice group. The background, which contains the non-ROI macroblocks, belongs to the last slice group. Every rectangular slice group is defined by the top

left and bottom right macroblock (see Fig. 1), and these two numbers are coded in a PPS instead of the complete MBAmap.

However, extracting of the ROI in the single-layered H.264/AVC bitstreams is almost impossible because of the following reasons:

- The ROIs can be selected by removing the background slice group from the bitstream. However, this results in an adapted bitstream that is not compliant anymore with the specification.
- Instead of removing the complete background, it can be replaced with other data. In [3], the authors discuss such a method in which the background slice group is substituted for placeholder slices. However, this introduces another drawback: the substitution gives rise to a drift effect in the decoded sequence because of a mismatch between the reference frames in the encoder and in the decoder.
- ROI scalability in H.264/AVC is just a coarse grain scalability technology. During the adaptation step, only a very limited (discrete) number of bit rates can be obtained.

In this paper, we use the scalable extension of H.264/AVC (i.e., SVC) in order to avoid the above-mentioned shortcomings.

An SVC bitstream can contain three embedded scalability axes at the same time (i.e., the temporal, spatial, and SNR axis). By combining these three scalability properties together with the FMO tool, ROI scalability can be obtained in an efficient manner. Our proposed combined scalability approach is given in Fig. 2 and details are provided below:

1. The original input sequence is down-sampled to obtain two spatial layers, i.e., the spatial scalability. Because both layers contain a lot of redundancy, the spatial base layer can be used as a prediction for the second spatial layer. In the spatial base layer, we do not define ROIs. Consequently, the FMO tool is disabled for the base layer. On the other hand, in the spatial enhancement layer, we have used FMO in order to define the ROIs of the sequence.
2. In each spatial layer, a temporal decomposition is executed in order the remove the temporal redundancy. For this purpose, hierarchical B pictures are used to obtain a pyramid structure as shown in Fig. 2.
3. In the second spatial layer (in which the ROIs are defined), each frame is extended with three Fine Grain Scalability (FGS) enhancement layers. These layers progressively increase the quality of the decoded frames. Because FMO is enabled in this spatial layer, for each slice in a slice group, a separate FGS NAL unit is coded. So, ROI scalability can be obtained by removing the FGS enhancement layers for the background slices and keeping the FGS layers for the ROIs. This results in ROIs having a better visual quality than the surrounding background. Because the quality base layer of each frame is always transmitted to the decoder, the adapted bitstreams (after the exploitation of the ROI scalability) are still compliant with the SVC specification. Furthermore, by using FGS enhancement layers, almost all bit rates can be obtained because FGS layers can be truncated at any byte position. As such, an optimal use of the capabilities of the usage environment can be obtained.
4. In order to realize the temporal decomposition, Motion Estimation/Motion Compensation (ME/MC) is done at the encoder and decoder side. Therefore, the reference frames have to be the same at the encoder and the decoder. However, the highest coding efficiency can only be obtained if the reference frame with the highest available quality (i.e., the frames decoded with 3 FGS layers) is employed for the ME/MC. However, any loss or truncation of an FGS enhancement layer (which is the result of ROI extraction or adjustment of the bit rate to satisfy constraints of the usage environment) results in a drift. To limit this drift effect after the adaptation, the ME/MC in the temporal base layer only makes use of the frames in the quality base layer and ignores the FGS layers. In all higher temporal layers, the FGS enhancement layers of the reference frames are also taken into account during the ME/MC step (see also Fig. 2). As such, the frames in the temporal base layer can be seen as re-synchronization points between the encoder and the decoder, resulting in a drift that is limited to the Group Of Pictures (GOP).

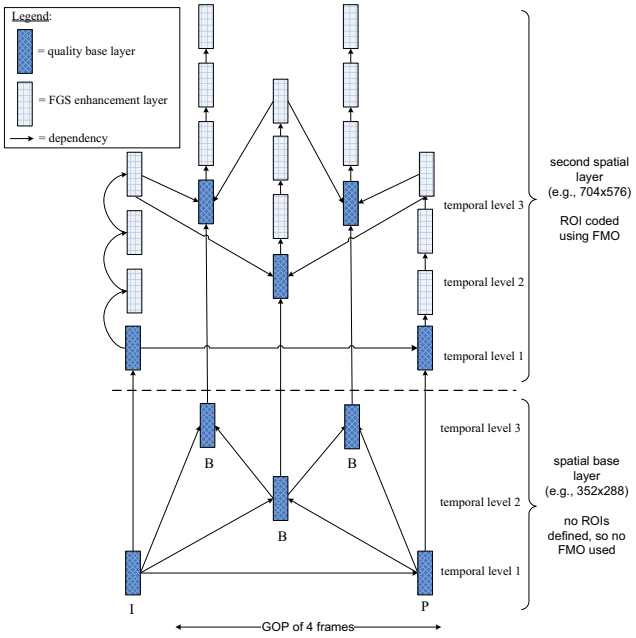

Fig. 2. ROI-based scalable coding pattern

To summarize, our approach towards ROI scalability results in bitstreams compliant with the specification (before and after the ROI extraction), in a limitation to inside the GOPs of the drift after the adaptation process, and in achieving a broad range of bit rates.

III. XML-BASED ROI EXTRACTION

A. MPEG-21 BSDL

MPEG-21 describes a multimedia framework that aims to enable a transparent and augmented use of multimedia resources across a wide range of networks and devices [4]. In such a framework, it should be possible to deliver scalable video content without the need to have knowledge regarding the underlying coding format.

Part 7 of the MPEG-21 Multimedia Framework, the Digital Item Adaptation (DIA, [5]) specification, offers a way to realize this goal. This solution relies on automatically generated XML-based descriptions containing information about the high-level bitstream structure. The DIA standard specifies MPEG-21 BSDL in order to describe the high-level structure of an encoded bitstream in XML. The generated documents are called Bitstream Syntax Descriptions (BSDs).
The entire chain of processes needed by the BSDL framework is given in Fig. 3. Explanatory notes are provided below:

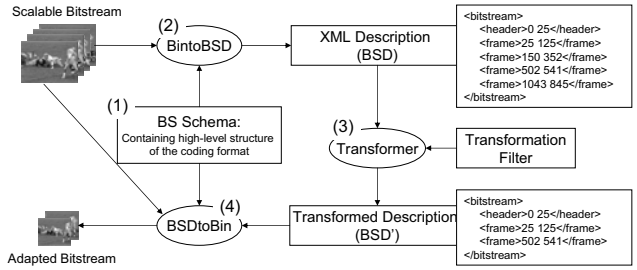

Fig. 3. The MPEG-21 BSDL framework

(1) The high-level structure of the coding format used is represented by a Bitstream Syntax Schema (BS Schema). The language used to construct such a BS Schema is BSDL, and this language is standardized in the DIA specification.
(2) The XML-based BSD of the scalable bitstream is generated by a format-agnostic parser once the original (scalable) bitstream and a corresponding BS Schema are known. The functioning of this parser, i.e., the BintoBSD Parser, is also defined in the standard.
(3) The generated description (i.e., the BSD) can subsequently be transformed by using a well-known XML transformation technology, e.g., eXtensible Stylesheet Language Transformations (XSLT, [6]). This transformation represents the adaptation process in the XML domain.
(4) The format-agnostic BSDtoBin Parser creates an adapted bitstream, using the transformed BSD, a corresponding BS Schema, and the original bitstream.

B. Adaptation in the XML domain

In this paper, we translate the high-level structure of SVC bitstreams into an XML document using BSDL [7]. The following syntactical data structures are described in XML: Supplemental Enhancement Information (SEI) messages, Sequence Parameter Sets (SPSs), PPSs, NAL unit headers, and the first three syntax elements of the slices. This information is needed to exploit the ROI scalability. In Fig. 4, a BSD fragment is given, representing a slice belonging to the second spatial layer (line 13) and quality base layer (line 14). Each NAL unit of the bitstream is described in XML in a similar manner. Based on the available information in the BSD, the extraction of the ROIs in the XML domain consists of the following steps:
Step 1: The slice groups are defined in the PPS. Based on this information, the MBAmap can be calculated and

```
1  <byte_stream_nal_unit>
     <zero_byte>0</zero_byte>
     <start_code_prefix_one_3bytes>000001</start_code_prefix_one_3bytes>
     <nal_unit>
5      <forbidden_zero_bit>0</forbidden_zero_bit>
       <nal_ref_idc>3</nal_ref_idc>
       <nal_unit_type>20</nal_unit_type>
       <nal_unit_information_for_scalable_extension>
         <simple_priority_id>0</simple_priority_id>
10       <discardable_flag>0</discardable_flag>
         <reserved_zero_bit>0</reserved_zero_bit>
         <temporal_level>0</temporal_level>
         <dependency_id>1</dependency_id>
         <quality_level>0</quality_level>
15     </nal_unit_information_for_scalable_extension>
       <raw_byte_sequence_payload>
         <coded_slice_of_a_non_IDR_picture_in_scalable_extension>
           <slice_layer_in_scalable_extension_rbsp>
             <slice_in_scalable_extension>
20             <first_mb_in_slice>198</first_mb_in_slice>
               <slice_type>0</slice_type>
               <pic_parameter_set_id>1</pic_parameter_set_id>
               <bit_stuffing>0</bit_stuffing>
               <slice_payload>99686 120</slice_payload>
25           </slice_in_scalable_extension>
           </slice_layer_in_scalable_extension_rbsp>
         </coded_slice_of_a_non_IDR_picture_in_scalable_extension>
       </raw_byte_sequence_payload>
     </nal_unit>
30 </byte_stream_nal_unit>
```

Fig. 4. Fragment of the BSD

the adaptation engine can localize the ROIs. However, it is possible that multiple PPSs are available in the bitstream. As such, the current PPS is activated by the PPS-id as indicated in the slice headers (see line 22).
Step 2: Once the MBAmaps are determined, the actual ROIs can be extracted. Based on the NAL unit type (line 7) and the dependency_id syntax element (line 13), one can determine to which spatial layer the current NAL unit belongs. So, XML fragments representing a NAL unit of the spatial base layer are copied without modification to the transformed BSD (the spatial base layer will not be removed).
Step 3: The remaining NAL units belong to the second spatial layer. The NAL units of the quality base layer (quality_id = 0, line 14) are also copied immediately to the transformed BSD. For the NAL units that are part of the FGS enhancement layers, one has to investigate whether the slice belongs to the ROI or not. Therefore, the first_mb_in_slice syntax element (line 20) is used to verify the location of the slice and to investigate to which slice group it belongs, using the MBAmap as determined in step 1. If the slice is a part of the background slice group, then the complete XML fragment is removed from the BSD. In the other case, the FGS layer can be truncated in order to satisfy the bit rate (e.g., substitute <slice_payload>99686 120</slice_payload> for <slice_payload>99686 60</slice_payload>).

IV. Performance results

To evaluate the performance of our XML-driven adaptation approach for the exploitation of ROI scalability, we have generated two scalable bitstreams. These bitstreams satisfy the scalable coding pattern that is visualized in Fig. 2 (in which the ROI is defined in the second spatial layer). The two sequences used are the well-known *Crew* and *Ice* sequences, having a resolution of 704×576 at the second spatial layer and a frame rate of 30Hz. The spatial and quality base layer are coded with a quantization parameter of 45.

TABLE I

RESULTS OF THE PERFORMANCE ANALYSIS

Description	Crew sequence	Ice sequence
#frames	400	230
Bit rate FMO disabled (kb/s)	2931.60	1579.27
PSNR FMO disabled (dB)	38.03	40.94
Bit rate FMO enabled (kb/s)	2955.14	1604.64
PSNR FMO enabled (dB)	38.04	40.93
PSNR ROI only (dB)	37.29	39.81
BintoBSD Parser (s)	19.07	11.05
Bit rate BSD (kb/s)	3349.94	3323.97
Bit rate compressed BSD (kb/s)	41.26	41.54
XSLT Transformation (s)	1.44	0.88
Bit rate transformed BSD (kb/s)	2283.59	2277.65
Bit rate compressed transformed BSD (kb/s)	29.41	30.26
BSDtoBin Parser (s)	3.87	2.68
Bit rate adapted bitstream (kb/s)	651,85	687.98
PSNR complete sequence	31.41	34.64
PSNR ROI only (dB)	36.21	39.24

The results of our measurements for the different engines (i.e., the BintoBSD Parser, the BSD transformer, and the BSDtoBin Parser) are given in Table I. The table is split up into three main parts: the first part contains information about the original scalable bitstreams; the second part evaluates our XML-driven adaptation framework; and finally in the third part, the characteristics of the adapted bitstreams (in which the ROIs are extracted) are given. From the table, one can see that both sequences give rise to the same conclusions.

We have coded the sequences once with and once without the FMO tool enabled. In case the FMO tool is disabled, the bitstreams do not contain a ROI. The bitstreams in which the FMO tool is enabled contain our predefined ROIs. These bitstreams are further used in our adaptation framework. However, one can observe that the overhead of using FMO is very limited (approximately 2%) while the same quality is kept. We have also made a distinction between the quality of the complete sequence on the one hand, and of the sequence only containing the ROI on the other hand. The latter is necessary to make a better comparison between the adapted bitstreams (in which the ROIs are extracted) and the original coded ones.

The BSD generation process (i.e., the BintoBSD Parser) is by far the most time-consuming part of our adaptation framework. Nevertheless, the execution time is acceptable in comparison with the encoding time (multiple hours). Moreover, this process should be executed only once. The generated BSDs are verbose: they are larger than the bitstream itself. However, by compressing these XML documents (e.g., by using WinRAR), the overhead of the XML documents is no longer a bottleneck for our framework. The following step, i.e., the transformation using XSLT, can be executed relatively fast. This process is the actual adaptation step in the XML domain, and it extracts the ROIs by removing all FGS layers of the background. The sizes of the transformed BSDs (in kb/s) are approximately a third of the original BSDs. This is because for each frame, 3 of the 9 available NAL units are removed. Finally, the generation of the adapted bitstream from the transformed BSD is very fast (i.e., the BSDtoBin Parser). The complete adaptation step (i.e., BSD transformation plus BSDtoBin Parser) can be done in real time.

In the last part of Table I, one can see that the bit rate of the adapted bitstreams, in which the ROIs are extracted, decreases significantly. The decrease in the bit rate results, of course, in an enormous drop of the quality of the complete sequence. However, the ROI still contains a similar quality as in the original coded bitstreams.

V. CONCLUSIONS

In this paper, we have explained how ROI scalability can be obtained in the Scalable extension of H.264/AVC (SVC). The ROIs are defined during the encoding phase of a video sequence by making use of FMO type 2. We have proposed a scalable coding pattern such that the ROIs can be extracted in an efficient manner. The adapted bitstreams (in which the ROIs are extracted) are compliant with the SVC specification. The ROIs in the adapted bitstream have a similar quality as in the original bitstream and there is no drifting effect in the decoded sequences. Moreover, a broad range of bit rates can be obtained. We have also proposed to do the adaptation process in a format-agnostic manner. Therefore, we have made use of MPEG-21 BSDL in order to describe the scalable bitstreams in XML. As a consequence, the extraction of the ROIs (i.e., the adaptation process) is executed in the XML domain instead of on the bitstream itself. The complete XML-driven adaptation framework for ROI scalability is evaluated in this paper. From this evaluation, we can conclude that the ROIs can be extracted in real time in the XML domain. Further, the background of the adapted sequences are characterized by a decreased quality, while the ROIs are still intact.

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] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Scalable Video Coding - Joint Draft 6," *Doc. JVT-S201*, April 2006.

[2] 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, June 2005.

[3] P. Lambert, D. De Schrijver, D. Van Deursen, W. De Neve, Y. Dhondt, and R. Van de Walle, "A real-time content adaptation framework for exploiting ROI scalability in H.264/AVC," in *Lecture Notes in Computer Science*, vol. 4179, September 2006, pp. 442–453.

[4] I. S. Burnett, F. Pereira, R. Van de Walle, and R. Koenen, *The MPEG-21 Book*. Wiley, John & Sons, Inc, 2006.

[5] "ISO/IEC 21000-7:2004 Information technology – Multimedia framework (MPEG-21) – Part 7: Digital Item Adaptation."

[6] M. Kay, *XSLT Programmer's Reference, 2nd edition*. Birmingham, UK: Wrox Press Ltd., 2001.

[7] 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 2006 IEEE ISCAS*, Island of Kos, Greece, May 2006, pp. 465–468.