A LOW-COMPLEXITY APPROACH FOR INCREASING THE GRANULARITY OF PACKET-BASED FIDELITY SCALABILITY IN SCALABLE VIDEO CODING

Heiner Kirchhoffer, Detlev Marpe, Heiko Schwarz, and Thomas Wiegand

Image Communication Group, Image Processing Department,
Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute
Einsteinufer37, D-10587 Berlin, Germany, [kirchhof|marpe|hschwarz|wiegand]@hhi.fraunhofer.de

ABSTRACT

Packet-based fidelity scalability (PFS) is a desirable feature in many video coding or transmission applications. Any realization of PFS in a hybrid video coding approach, however, requires suitable concepts for controlling drift and for generating sufficiently small increments in bit rate in order to allow progressive refinements of perceptual quality relative to a given base layer quality. This paper addresses those problems in the context of the scalable video coding (SVC) extension of H.264/AVC. We present an algorithmically simple but yet remarkably well-performing method for packet-based fidelity scalability that is maximally consistent with the existing entropy coding design of H.264/AVC, allows sufficiently small increments in bit rate, and has been adopted as a normative element of SVC. We also discuss the benefits of the key picture concept of SVC in view of our proposed PFS approach. Experimental results are presented that demonstrate the effectiveness of our method for a few selected SVC conforming encoder configurations.

Index Terms— SVC, H.264/AVC, fidelity scalability, CABAC

1. INTRODUCTION

In the Joint Draft (JD 9) [2] of the scalable video coding (SVC) extension of H.264/AVC [1] two methods of fidelity scalability are specified, so-called *coarse granular* or *layer-based scalability* (LFS) and *medium granular* or *packet-based scalability* (PFS). Conceptually, LFS is based on the multi-layer concept of SVC, which means that usually only a small number of discrete fidelity layers is supported. As an additional restriction, the configuration of fidelity layers in terms of supported target rate-distortion (R-D) points or potential switching points between them must be determined at the encoding time and is therefore fixed in advance. This may not be suitable for certain applications where, e.g., a higher flexibility in terms of bitstream adaptation is requested. Consequently, as a variation of LFS, PFS has been included in the SVC design, which allows switching between different fidelity layers at virtually any arbitrary point in the bitstream. However, as an important side condition for PFS, so-called *key pictures* [3] have to be inserted as re-synchronization points on a regular basis in order to limit the drift effects that may result from discarding fidelity refinement packets which have been included as a reference for motion-compensated prediction at the encoder.

When using inter-layer residual prediction in SVC, both LFS and PFS are typically realized by generating a fidelity scalable representation of the residual texture signal. This scalable representation is conceptually composed of a *base layer* and one or several *enhancement layers*. The base-layer residual texture signal is obtained by first quantizing the residual texture signal with a certain pre-determined quantization step size and corresponds to a reconstruction with the lowest supported quality. The additional quality enhancement layers on top of the base layer are generated by repeatedly re-quantizing the quantization error that results from a possible change in motion parameters and from the quantization of the base layer and, if more than one enhancement layer is involved, all preceding enhancement layers. Typically, the quantization step size is halved from one layer to the next in the corresponding encoding process in order to limit the bit rate overhead relative to single-layer coding. At the decoder, the received number of fidelity enhancement layers is added to the base layer in order to create a reconstruction of the residual texture signal. In this way, each of the fidelity enhancement layers represents a discrete step of quality improvement which has to be applied fully or not at all. But in some cases it could be desirable to scale the fidelity of the video signal in smaller steps than by using a whole fidelity enhancement layer.

The initial design of SVC [3] included a method for achieving *fine granular scalability* (FGS), which allowed a truncation of fidelity enhancement packets in a way that virtually any byte could be used for progressively refining the residual texture signal. This FGS method, though quite competitive in terms of R-D performance compared to single-layer coding, has the disadvantage of being computationally quite demanding due to its related multi-pass entropy coding stage. In this paper, we present an approach for increasing the granularity of the packet-based fidelity scalability method in SVC. Similar to the FGS design, it operates in the transform domain and allows fragmentation of a given fidelity enhancement by means of frequency-selective grouping of transform coefficients. However, as an important distinct feature, the bitstream syntax and the entropy coding design of H.264/AVC has been re-used to a maximum extent. The degree of fragmentation can be chosen by the encoder in a flexible way without significantly compromising R-D performance.

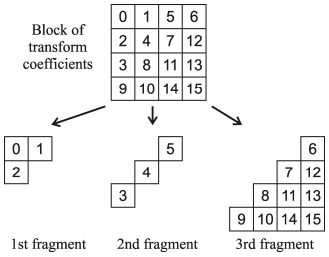

Fig. 1 – Fragmentation example for a 4×4-block of transform coefficients.

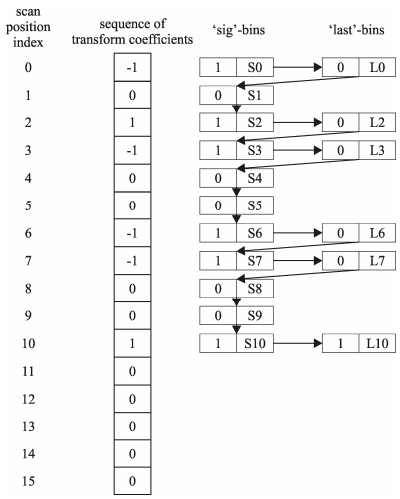

Fig. 2 –Encoding of a sample significance map in H.264/AVC CABAC for a single layer without fragmentation.

2. SCHEME FOR INCREASING THE GRANULARITY OF FIDELITY SCALABILITY

Residual texture signals in SVC are represented in the form of 4x4- and/or 8x8-blocks of transform coefficients. These blocks are converted into sequences of transform coefficients by traversing them using the so-called zig-zag scan. The basic idea of our proposed fragmentation scheme is to split the sequences of transform coefficients into a certain number of fragments, for each block in the same way. In this way, a separate sub-layer is generated for each of the resulting fragments. The first sub-layer then contains the first fragment of each block, the second sub-layer contains the second fragment of each block, and so on. An example for such a fragmentation is depicted in Fig. 1 for the case of a 4x4 block which is split into three fragments. Each of these sub-layers is encoded in a separate slice with a *start index* and a *stop index* indicating the first and last index, respectively, in scan order that is included in the corresponding slice and for which those parameters are signaled in the slice header. For an 8x8 block, there is a separate zig-zag scan pattern and to avoid signaling a separate start and stop index for the 64 transform coefficients of 8x8 blocks, the start and stop index signaled for 4x4 blocks is each multiplied by 4 to derive a corresponding start and stop index for 8x8 blocks. In other words, each scan position of a 4x4 block is associated with 4 scan positions of an 8x8 block. For the chroma components as well as for the luma residual texture signal of the intra 16x16 prediction mode, a further transform stage is applied to the resulting DC coefficients, according to H.264/AVC. To treat this fact properly, all DC coefficients are transmitted in the fragment that includes scan index 0.

Note that our approach does not distinguish between a full fidelity scalable layer and sub-layers. For each (sub-) layer in SVC, just a first and a last scan position is specified in the slice header. And accordingly, during the encoding or decoding of transform coefficients, only these scan positions are processed. In this way, a full fidelity scalable layer can be split into 2 to 16 sub-layers. These remarkably small changes are sufficient to achieve improved fidelity scalability on a packet/slice basis. A detailed description of the approach can be found in [6].

3. IMPACT ON RATE-DISTORTION PERFORMANCE

H.264/AVC specifies two different entropy coding schemes. The first one is called CAVLC (Context Adaptive Variable Length Codes) and is computationally less demanding but also less efficient than the second one, which is called CABAC (Context-based Adaptive Binary Arithmetic Coding). CABAC includes a flexible probability modeling apparatus and thus achieves a higher coding efficiency than CAVLC. Our proposed approach can be realized for both entropy coding schemes. However, in this paper, all observations are based on the CABAC entropy coding mode. A CAVLC-based entropy coding scheme for fragmented residual texture signals can be found in [5] (Test method 3).

In the case of CABAC, the encoding of sequence of transform coefficients in a given block consists of two parts. The first part is related to the encoding of the so-called significance map, indicating which transform coefficient levels in a block are non-zero. The second part, denoted as 'level coding' in this paper, deals with the values of all non-zero coefficient levels in a block. Note that with our proposed fragmentation scheme, we are re-using those CABAC parts without any changes. However, in order to find out the impact of the partitioning of transform coefficients with respect to CABAC processing, we will briefly discuss in the following how both coding parts are operating in the case of a fidelity enhancement layer with and without fragmentation.

In a fidelity scalable layer, most of the transform coefficient levels usually take one of the values -1, 0, or +1. This is due to the fact that a residual texture signal inside a fidelity enhancement layer mainly represents a re-quantized quantization error and that re-quantization is usually done by halving the quantization step-size. Higher absolute values for some of the transform coefficient levels may result from motion parameter refinements or from using smaller quantization step sizes in fidelity enhancements. For the CABAC level coding part, this means that in most cases only absolute values of 1 plus all related signs have to be coded. Since coding of signs in CABAC is based on a fixed probability model, fragmentation may

only affect coding of the absolute level values. However, assuming the dominant case of a single possible non-vanishing absolute value of 1 and further neglecting the adaptation phase of the corresponding probability models, each of the level values can be assumed to be encoded with the minimum possible codelength in CABAC, regardless of the chosen fragmentation.

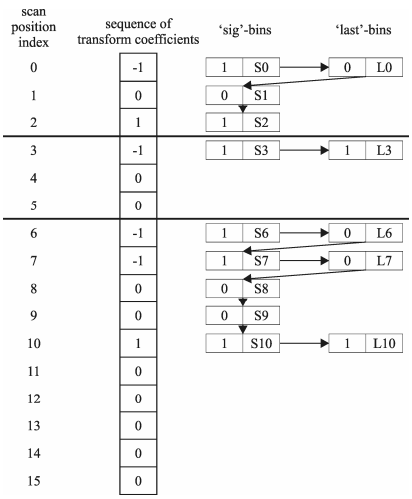

Fig. 3 – Encoding of a sample significance map in H.264/AVC CABAC by using multiple fragments.

For the significance coding part of CABAC, Fig. 2 and Fig. 3 illustrate the processing of a sample significance map for the non-fragmented and the fragmented case, respectively. In general, CABAC encoding proceeds in a way that a sequence of so-called bins is derived, where each bin consists of a bit that carries the information to be signaled in the bitstream. Furthermore, each bin is associated with one probability model and to simplify matters, it can be assumed that CABAC encodes all bins related to a certain probability model with the entropy rate calculated by including all bins encoded with this model. Each bin in Figs. 2 and 3 is depicted as two joint boxes with the value of the bit indicated in the left box and the identifier of the probability model in the right box. The arrows indicate the coding order. For each non-zero coefficient, a 'sig' bin with value 1 is encoded, and afterwards a so-called 'last' bin. The 'last' bin has a value of 1, if there is no non-zero coefficient left for this block, and it has a value of 0, otherwise. If the 'last' bin is 1, all following 'sig' bins would have values equal to 0. Hence, the 'last' bins serve the purpose of avoiding a sequence of trailing zero-valued 'sig' bins to be signaled [1].

The bins that are generated in the fragmented case are depicted in Fig. 3 for the fragmentation example of Fig. 1. Exactly the same number of 'sig' bins with values of 1 as in the non-fragmented case is coded. However, some 'last' bins toggle from a value of 0 to 1 and hence, some 'sig' bins with values equal to 0 are not coded at all. Additionally, each 'last' bin at the last scan index of a given fragment is not needed anymore since there are no scan positions left to which the 'last' bin would relate to. In principle, these changes do have an impact on the bit rate, but it is not *a priori* clear whether the overall bit rate increases or decreases. As shown in Sec. 5, the overall impact on bit rate is rather small in most observed cases.

4. DRIFT CONTROL

If a block uses motion-compensated prediction (MCP), a displaced block of a previously-coded frame is used as prediction. In a non-scalable scenario, it has to be ensured that encoder and decoder use exactly the same predicted block which is known as closed-loop processing. But in a fidelity scalable scenario with packet-based granularity, a given frame may be of varying picture quality depending on how many fidelity scalable packets have been included at the decoder.

Drift between encoder and decoder reconstructions can only be completely avoided when the base-layer reconstruction is always used for MCP of following pictures. This has the disadvantage that the bit rate that is spent for encoding fidelity enhancements of a picture cannot be employed for fidelity enhancement of following pictures. Furthermore, two representations for each picture have to be decoded: A representation without fidelity enhancements that is used for MCP of following pictures, and a representation with fidelity enhancements that is output.

Another solution is to allow different predictions at encoder and decoder and thus introduce drift. If this drift is carefully controlled, a higher rate-distortion performance can be achieved in comparison to the case that does not allow drift. In SVC, the above described concept for completely avoiding drift is only used in so-called key pictures, which are usually regularly inserted in a bitstream and serve as re-synchronization points between encoder and decoder. For all other pictures, the representations of the reference pictures with the highest available fidelity are used for MCP. When combining this concept with hierarchical prediction structures, drift can be efficiently controlled while achieving a coding efficiency close to single-layer coding. More details about the key picture concept and how it can be efficiently combined with hierarchical prediction structures can be found in [4].

A partitioning of the residual texture data of a fidelity scalable layer fits seamlessly into this concept. The drift control concept for packet-based fidelity scalability does not require any modification for the presented approach of increasing the granularity for PFS.

5. EXPERIMENTAL RESULTS

Three sample configurations of fragments have been investigated. The first and second configuration consist of three fragments each, containing 3, 3, and 10 scan positions (first), and 2, 2, and 12 scan positions (second), respectively, while the third configuration consists of two fragments with 6 and 10 scan positions.

Each of the R-D plots in Fig. 4 shows four R-D curves. The curve denoted as "not fragmented" consists of two R-D points, where the lower left point corresponds to the decoded base layer only and the upper right point consists of one additional SVC conforming fidelity enhancement layer without any fragmentation.

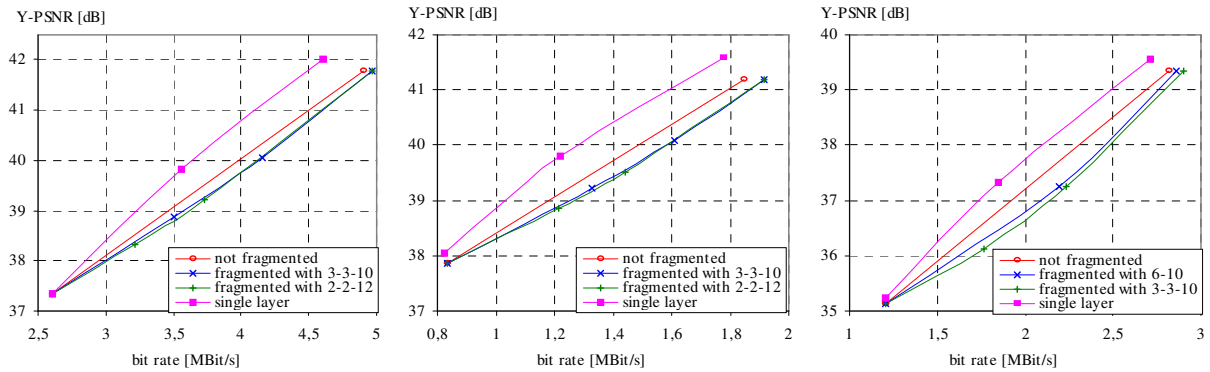

Fig. 4: – R-D plots for sequence "Football" (left, intra-only), "Crew" (middle, GOP 16), and "Harbour" (right, GOP 16), CIF at 30 Hz.

Each point of the R-D curve denoted as "single layer" corresponds to the non-scalable case of separately encoded H.264/AVC-conforming bitstreams. Each of the two remaining curves shows the R-D performance of the proposed fragmented SVC bitstreams with one or two additional R-D points as compared with the non-fragmented case. These additional R-D points were generated by simply discarding one or two upper sub-layers uniformly for each picture of the corresponding test sequence. Fig. 4 shows results for the intra-only case (left) and the case of using 16 hierarchical B-pictures per group of pictures (GOP 16: middle and right). As can be seen from the corresponding R-D graphs, the fragmentation has only a minor negative impact on the R-D performance at the highest rate points. Note that the interior points belonging to the various fragmented R-D curves should not be compared to the 'not fragmented' R-D curve, since the latter is just a straight line drawn between the base layer R-D point and the fidelity enhancement layer R-D point.

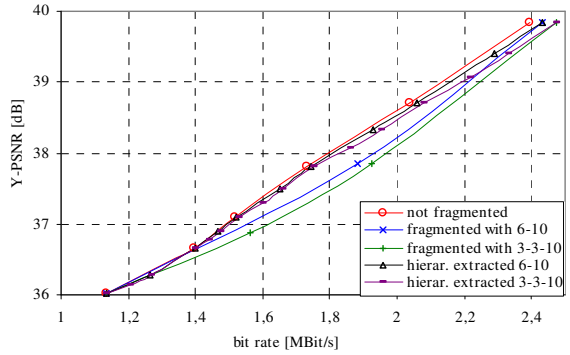

Fig. 5 – R-D plot for sequence "Bus" (CIF at 30 Hz).

In case of using hierarchical B-pictures, a further improvement in terms of granularity can be achieved by selectively discarding fidelity enhancement packets from pictures of each temporal hierarchy level. A corresponding result for such a case is shown in Fig. 5, where R-D curves of bitstreams are plotted that were generated by using a dyadic hierarchy of 5 temporal levels (GOP 16). As before, the lower left and upper right R-D point of each curve corresponds to the base layer and full fidelity enhancement layer, respectively. However, in the non-fragmented case, 4 additional extraction points are generated by incrementally discarding packets of the highest temporal level, the two highest levels, the three highest levels, and so on. For the fragmented cases, extraction points are generated in the same way and on top of this, only some of the available fragments of the highest retained temporal level may be discarded. This leads to one (6-10) or two (3-3-10) additional extraction points between each two non-fragmented rate points. Compared to the simple case of uniformly discarding packets across all temporal layers (denoted as "fragmented"), this selective extraction along the temporal hierarchy (denoted as "hierar. extracted") results in a distinctly improved R-D performance at intermediate rate points, as can be seen in Fig. 5. Note however, that our additionally investigated extraction scheme is not the only possible alternative for selectively extracting fidelity scalable packets from of a given SVC bitstream with hierarchical B-pictures. Application of more sophisticated extraction rules may further improve the R-D performance of PFS in SVC.

6. CONCLUSION

A low-complexity scheme for increasing the granularity of packet-based fidelity scalable coding in SVC has been presented. It was shown that an existing fidelity scalable layer can be split into several independently decodable layers without significantly increasing the overall bit rate.

REFERENCES

[1] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 8 (including SVC extension): consented in July 2007.

[2] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, M. Wien, "Joint Draft 9 of SVC Amendment," Joint Video Team Doc. JVT-V201, Marrakech, Morocco, January 2007.

[3] H. Schwarz, T. Hinz, H. Kirchhoffer, D. Marpe, and T. Wiegand, "Technical description of the HHI proposal for SVC CE1," ISO/IEC JTC 1/SC29/WG11, Doc. M11244, Palma de Mallorca, Spain, Oct. 2004.

[4] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuit and Systems for Video Technology,* vol. 17, no. 9, Sept. 2007.

[5] J. Ridge, "CE1: FGS Simplification," Joint Video Team Doc. JVT-V301r1, Marrakech, Morocco, January 2007.

[6] H. Kirchhoffer, H. Schwarz, T. Wiegand, "CE1: Simplified FGS," Joint Video Team Doc. JVT-W090, San Jose, CA, USA, April 2007.