On the impact of the GOP size in a temporal H.264/AVC-to-SVC transcoder in Baseline and Main Profile

Rosario Garrido-Cantos, Jan De Cock, José Luis Martinez, Sebastiaan Van Leuven, Pedro Cuenca, Antonio Garrido, and Rik Van de Walle

In: Springer Multimedia Systems, 19 (2), 163-177, 2013.

http://link.springer.com/article/10.1007/s00530-012-0287-8

On the Impact of the GOP Size in a Temporal H.264/AVC-to-SVC Transcoder in Baseline and Main Profile

R. Garrido-Cantos · J. De Cock · J.L. Martínez ·
S. Van Leuven · P.Cuenca· A. Garrido ·
R. Van de Walle

Abstract: Scalable Video Coding is a recent extension of the Advanced Video Coding H.264/AVC standard developed jointly by ISO/IEC and ITU-T, which allows adapting the bitstream easily by dropping parts of it named layers. This adaptation makes it possible for a single bitstream to meet the requirements for reliable delivery of video to diverse clients over heterogeneous networks using temporal, spatial or quality scalability, combined or separately. Since the Scalable Video Coding design requires scalability to be provided at the encoder side, existing content cannot benefit from it. Efficient techniques for converting contents without scalability to a scalable format are desirable.

In this paper, an approach for temporal scalability transcoding from H.264/AVC to Scalable Video Coding in Baseline and Main Profile is presented and the impact of the GOP size is analyzed. Independently of the GOP size chosen, time savings of around 63% for Baseline Profile and 60% for Main Profile are achieved while maintaining the coding efficiency.

Keywords Multimedia System ・Scalable Video Coding (SVC) ・H.264/AVC ・Transcoding ・Temporal Scalability

R. Garrido-Cantos (✉)・P. Cuenca ・A. Garrido
Albacete Research Institute of Informatics, University of Castilla-La Mancha, Campus Universitario s/n, 02071 Albacete, Spain
e-mail: charo@dsi.uclm.es

P. Cuenca
e-mail: pcuenca@dsi.uclm.es

A. Garrido
e-mail: antonio@dsi.uclm.es

J. De Cock ・S. Van Leuven ・R. Van de Walle

Department of Electronics and Information Systems, Multimedia Lab, Ghent University–IBBT, Gaston Crommenlaan 8, bus 201, 9050 Ledeberg, Ghent, Belgium

J. De Cock
e-mail: jan.decock@ugent.be

S. Van Leuven
e-mail: sebastiaan.vanleuven@ugent.be

R. Van de Walle
e-mail:rik.vandewalle @ugent.be

J.L. Martínez
Architecture and Technology of Computing Systems Group, Complutense University, Ciudad Universitaria s/n, 28040 Madrid, Spain
e-mail: joseluis.martinez@fdi.ucm.es

1 Introduction

In the last years, the demand for multimedia applications and terminals to visualize these contents has increased. These multimedia contents, generally, are encoded to reduce the necessary storage capacity and the consumption of bandwidth.

Nowadays, the existing networks that transmit these contents are heterogeneous and the receiving devices have different decoding capacities (display, processing power, memory capabilities, etc). An encoded video stream typically only meets specific requirements of frame rate, resolution and SNR (quality), so it cannot be adapted to this variety of bandwidth constraints and terminals [1]. Scalable codecs, however, allow different types of video adaptation [2][3].

Recently, the scalable extension of H.264/AVC [4] was standardized. Scalable Video Coding (SVC) [4] provides temporal, spatial, quality scalability or a combination of these. The SVC bitstream is organized in layers (one base layer and one or more enhancement layers). The base layer represents the lowest frame rate, spatial resolution and quality resolution while the enhancement layers provide improvements allowing a higher frame rate, resolutions and/or quality. The bitstream is adaptable to the channel bandwidth or the terminal capabilities by truncating the undesirable enhancement layers.

Today, most of the video contents are still created in a single-layer format (H.264/AVC video streams) so they cannot benefit from this scalability. This fact leads to requirements for developing alternative techniques to enable video adaptation between non-scalable and scalable bitstreams. In this paper, an efficient video transcoding [5] technique is proposed for transforming H.264/AVC bitstreams to SVC bitstreams providing temporal scalability. Its efficiency is obtained by reusing as much information as possible from the original bitstream, such as motion information. The ultimate goal is to perform the required adaptation process faster than the straightforward concatenation of decoder and encoder while maintaining the coding efficiency. This technique is applied to different sequences using varying GOP sizes in Baseline and Main Profile. This paper also analyzes the impact of this size in the behavior of the proposal.

The remainder of this paper is organized as follows. Section 2 describes the related work for H.264/AVC to SVC transcoding at this moment. In Section 3, a brief introduction to SVC is given. In Section 4, our approach is described and in Section 5 results of applying our approach to different combinations of GOP sizes are presented. Finally, conclusions are shown in Section 6.

2 Related Work

In the literature different techniques exist for homogeneous transcoding such as [6][7] and for heterogeneous transcoding [8][9]. In the framework of SVC-based video transcoding, several techniques have been proposed in the recent past for introducing scalability in compressed bitstreams. Most of these proposals are related to quality-SNR scalability, although there are some proposals for spatial and temporal scalability.

For quality-SNR scalability, the first work in this field was done in 2006 by Shen et al. They proposed a mode decision method in SVC domain for transcoding from hierarchically encoded H.264/AVC to Fine-Grain Scalability (FGS) streams [10]. In 2009, De Cock et al. presented different architectures with different rate distribution flexibility and computational complexity for transcoding from single layer H.264/AVC bitstream to SNR scalable SVC streams with Coarse-Grain Scalability (CGS) layers [11].

Regarding spatial scalability, in 2009, Sachdeva et al. proposed a transcoder from single-layered H.264/AVC to multi-layered SVC which allows adding spatial scalability to existing non-scalable H.264/AVC video streams. The algorithm reuses available data by efficient downscaling of video information for different layers. The idea consists of an information downscaling algorithm which uses the top enhancement layer (which has the same resolution as original) by means of bypassing various time consuming processes of the encoding algorithm [12].

Finally, for temporal scalability, in 2008, Dziri et al. presented a transcoding method from H.264/AVC P-picture based bitstreams to an SVC bitstreams with temporal scalability [13]. In 2010, another technique for transcoding providing temporal scalability was proposed by Al-Muscati et al. This method was applied in Baseline Profile and reuses MB codings and derivate new MVs from the MVs from the H.264/AVC stream [14]. The same year, Garrido-Cantos et al. presented an H.264/AVC-to-SVC video transcoder that efficiently reuses some motion information of the H.264/AVC decoding process in order to reduce the time consumption of SVC encoding algorithm. The approach was developed for Main Profile and dynamically adapted for P and B frames with several temporal layers and can be used with different GOP sizes [15].

In this paper, we extend the analysis of the impact of the GOP size presented previously in [15] by adding an analysis for Baseline Profile of the impact of the number of temporal layers as well as an analysis of the impact of the GOP size.

3 Scalable Video Coding

Scalable Video Coding (SVC) was standardized in 2007 and is an extension of H.264/AVC. SVC streams are composed of layers which can be in order to adapt the streams to the needs of end users, the capabilities of the terminals or the network conditions.

The layers are divided into one base layer and one or more enhancement layers which employ data of lower layers for efficient coding.

SVC allows three types of scalability: temporal, spatial, quality scalability or a combination of these. The base layer represents the lowest frame rate, the lowest resolution and the lowest quality respectively.

SVC uses coding tools of H.264/AVC and additionally provides inter-layer prediction methods which allow an exploitation of the statistical dependencies between different layers for improving the coding efficiency of enhancement layers.

A simplified block diagram of the SVC coding structure is shown in Figure 1.

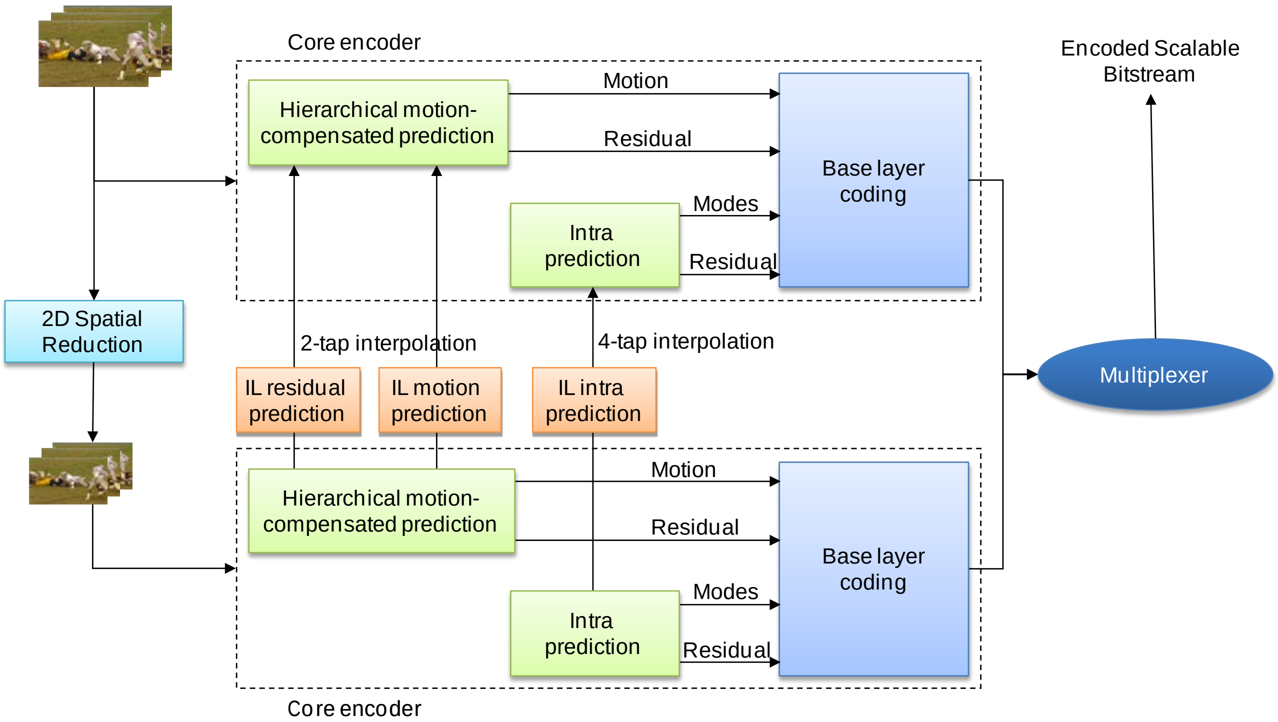

Fig. 1 Block diagram of a SVC encoder

In the following sections, the different scalabilities supported by SVC are explained with more detail. For a comprehensive overview of the scalable extension of H.264/AVC, the reader is referred to [16].

3.1 Temporal Scalability

A bitstream provides temporal scalability when it can be divided into a temporal base layer (with an identifier equal to 0) and one or more temporal enhancement layers (with identifiers that increase by 1 in every layer), so that if all the enhancement temporal layers with an identifier greater than one specific temporal layer are removed, the remaining temporal layers form another valid bitstream for the decoder.

In H.264/AVC and by extension in SVC, any picture can be marked as reference picture and used for motion-compensated prediction of following pictures. This feature allows coding of picture sequences with arbitrary temporal dependencies. In this way, to achieve temporal scalability, SVC links its reference and predicted frames using hierarchical prediction structures [17] which define the temporal layering of the final structure. In this type of prediction structures, the pictures of the temporal base layer are coded in regular intervals by using only previous pictures within the temporal base layer as references. The set of pictures between two successive pictures of the temporal base layer together with the succeeding base layer picture is known as a Group of Pictures (GOP). As it was mentioned previously, the temporal base layer represents the lowest frame rate that can be increasing by adding pictures of the enhancement layers.

There are different structures for enabling temporal scalability, but the typical GOP structure is based on hierarchical B pictures with a dyadic structure, which is also used by default in the Joint Scalable Video Model (JSVM) reference encoder software [18]. The number of temporal layers is thus equal to 1+ $\log_2$[GOP size]. One of these structures with a GOP of 8 (I7BP pattern) and therefore four temporal layers, is illustrated in Figure 2 where the temporal base layer is represented by TL0 and the successive temporal layers increase the identifier by 1.

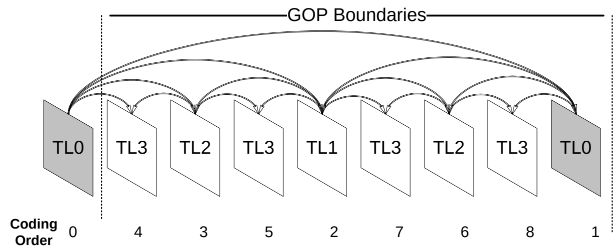

Fig. 1 Hierarchical B pictures with dyadic prediction structure containing four temporal layers (TL)

This structure provides another three independently decodable sub-sequences with 1/8, 1/4, and 1/2 of the full original frame rate.

3.2 Spatial Scalability

For supporting spatial scalable coding, SVC follows the approach of multi-layer coding. Each layer corresponds to a supported spatial resolution and is referred to by a spatial layer or dependency identifier. The layers are coding following an oversample pyramid for each resolution (QCIF, CIF, 4CIF, 16CIF). A multi-layer structure that enables spatial scalability is shown in Figure 3.

The pictures of different spatial layers are independently coded as for single-layer coding. However, in order to improve the coding efficiency of the enhancement layers, additional inter layer prediction mechanisms have been introduced. These mechanisms allow an exploitation of the statistical dependencies between different layers for improving the coding efficiency of enhancement layers. All these methods can be chosen on a macroblock or block basis allowing the encoder to select the coding mode that gives the highest coding efficiency.

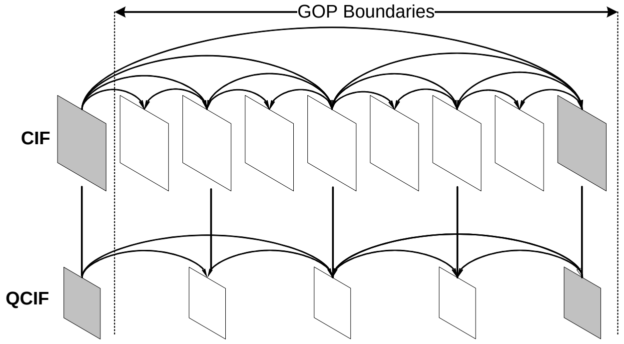

Fig. 2 Multilayer structure with addional inter layer prediction for enabling spatial scalable coding

3.3 Quality Scalability

Quality scalability in SVC (SNR scalability) is intended to provide different levels of quality to the original video. It can be seen as a case of spatial scalability where the base and enhancement layers have identical pictures sizes, but different qualities.

Different techniques for quality scalability have been designed: Coarse-Grain Scalability (CGS), Medium-grain Scalability (MGS) and Fine-Grain Scalability (FGS). A specific design for FGS was not included in the definitive version of SVC.

In CGS (Figure 4), quality scalability is obtained by varying quantization in the different layers. Firstly, the base layer, which is compatible with H.264/AVC, is generated. Then, for every quality layer added, the encoding techniques of H.264/AVC are combined with inter-layer prediction tools. This can make the compression of enhancement layers more efficient.

MGS (Figure 5) uses techniques similar to CGS, but provides more flexibility. It is obtained by dividing the quantized coefficients over different packets. MGS allows switching between different MGS layers and the key picture concept, which allows the adjustment of a suitable trade-off between drift and enhancement layer coding efficiency for hierarchical prediction structures.

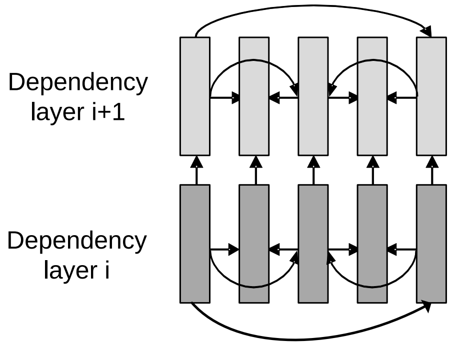

Fig. 3 Dependencies for CGS quality scalability based on dependency layers

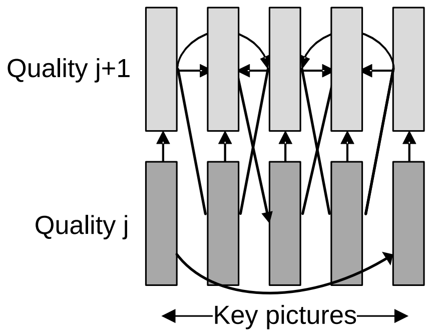

Fig. 4 Dependencies for MGS quality scalability based on dependency layers

4 H.264/AVC-to-SVC Transcoding

The most time consuming tasks carried out at H.264/AVC and SVC encoders are the Motion Estimation (ME) and the macroblock (MB) mode decision processes. Both techniques perform the motion-compensated prediction. Because of their high complexity, they are the most suitable modules to be accelerated. In this approach we are focusing on the ME process.

The ME process consists in eliminating the temporal redundancy in a way to determine the movement of the scene. For this purpose, Motion Vectors (MVs) between every block and a block of the reference frame are calculated. This block of the reference frame is the most similar inside the search area to the current block. The idea behind the proposed transcoder consists of reusing the MVs collected in the H.264/AVC decoding algorithm (as part of the transcoder) to accelerate the SVC ME process. These MVs generated by H.264/AVC represent, approximately, the amount of movement of the MB or sub-MB and can be reused to accelerate the SVC ME process by reducing the search area dynamically. A scheme of this approach is shown in Figure 6.

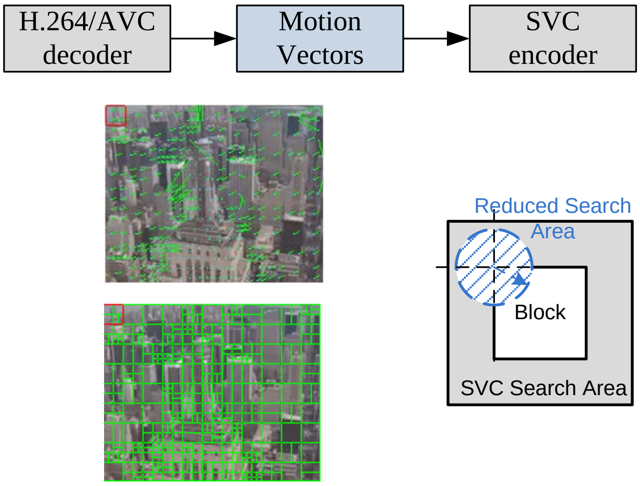

Fig. 5 Scheme of the proposed transcoder.

An initial search area, smaller than the SVC one, is determined by the circumference centered in (0,0) point for each MB or sub-MB. This circumference has a radius which varies dynamically depending on the incoming vectors for a specific MB. In H.264/AVC and SVC, there is a MV for each MB or sub-MB. As the MB partitioning can be different for the same picture in H.264/AVC and SVC (see Figure 7), it cannot exist a one-to-one mapping between previously calculated H.264/AVC MVs and the SVC MVs. To overcome that situation the radius of the new reduced search area will be the length of the average of the incoming vectors of a determined MB of H.264/AVC.

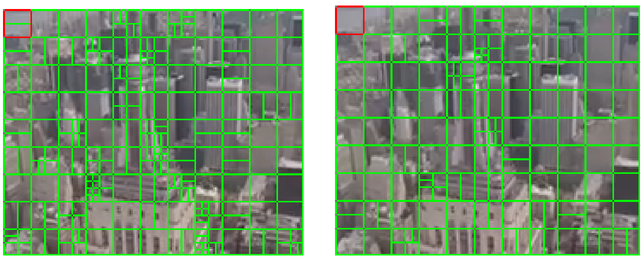

Fig. 6 MB mode decisions generated by H.264/AVC (left) and SVC (right) for the 4th frame in the City QCIF sequence

Another difficulty to overcome is the mismatch between GOP sizes, GOP patterns and prediction structures. While the starting encoded bitstream in H.264/AVC is formed by IBBP/IPPP GOP patterns (in Main and Baseline Profile respectively) without temporal scalability, the final SVC bitstream can benefit from the temporal scalability of hierarchical structures (see Figure 2). This fact leads to the possibility to have different MVs in both H.264/AVC and SVC.

In general, hierarchical GOP structures will cause motion-compensated prediction to use a longer distance between a frame and its reference. This distance increases when the identifier of the temporal layer decreases. To deal with this different prediction distance, a correction factor is introduced. This factor depends on which temporal layer the current frame is in. For building the final search area, the radius of the circumference generated previously is multiplied by this correction factor.

So in a formal way, the initial search window is limited by the area S defined as:

$$S = \{(x,y) \mid (x,y) \in (A \cap C)\}$$

where (x,y) are the coordinates to check, A is the search range used by SVC and C is the circumference which restricts the initial search area with centre on the upper left corner of the MB or sub-MB. C is the circumference defined by:

$$C^2 = r_x^2 + r_y^2$$

where r_x and r_y are calculated depending of the average of MVs of the H.264/AVC MB (MV_x and MV_y) or a minimum value is set to avoid applying too small search ranges.

$$r_x = \max(MV_x, \min\_value)$$

$$r_y = \max(MV_y, \min\_value)$$

When the initial search area is defined, it has to be adjusted by multiplying the coordinates of the radius by the correction factor, so the final radius would be defined as follows:

$$r_x = coef(n) \cdot \max(MV_x, \min\_value)$$

$$r_y = coef(n) \cdot \max(MV_y, \min\_value)$$

where *coef* depends on the identifier of the temporal layer (n) where the frame is in:

$$coef(n) = GOP_{length} / 2^n$$

This proposal is valid for Baseline and Main Profile, although for that one the possible different prediction lists have to be taken into account. H.264/AVC and SVC use two lists of previously-coded reference frames (list0 and list1), before or after the current picture in temporal order in B pictures for prediction. For P pictures only list0 is used. An example of prediction modes in B frames is shown in Figure 8.

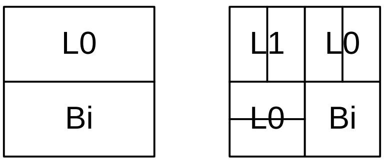

Fig. 7 Examples of prediction modes in B macroblocks (L0: list0, L1: list1, Bi: bi-predictive).

Due the different GOP patterns between H.264/AVC and SVC, it is usual to have cases where MVs extracted from H.264/AVC are obtained with a reference of a list0, but SVC needs there reference from the list1 or vice versa or even a bi-predictive prediction is done requiring MVs of both lists. In these cases, the supposition is made that the length of the MV of both lists for a MB is the same.

5 Implementation Results

In this section, results from the implementation of the technique described in the previous section are shown. Test sequences with varying characteristics were used, namely Hall, City, Foreman, Soccer, Harbour, Crew, Football and Mobile in CIF resolution (30 Hz) and QCIF resolution (15 Hz).

These sequences were encoded using the H.264/AVC Joint Model (JM) reference software, version 16.2 [19], with an IPPP pattern for Baseline Profile and an IBBP pattern for Main Profile with a fixed QP = 28 in a trade-off between quality and bitrate. Then, for reference results, encoded bitstreams were decoded and re-encoded using the JSVM software, version 9.19.3 [18] with temporal scalability and different values of QP (28, 32, 36, 40). For results of the technique, encoded bitstreams in H.264/AVC were transcoded using the technique described in Section 4. Different sets of GOP length were used for CIF and QCIF sequences to test the impact of the GOP size in our proposal.

The GOP sizes used are 2, 4, 8, 16, and 32 for QCIF resolution and 4, 8, 16, 32, and 64 for CIF resolution. These values of GOP sizes were chosen to have a base layer picture at the same time instances for QCIF and CIF resolutions as shown in Figure 9 and Figure 10. With these GOP sizes, these pictures are inserted every 0.13s, 0.26s, 0.5s, 1s, and 2s respectively.

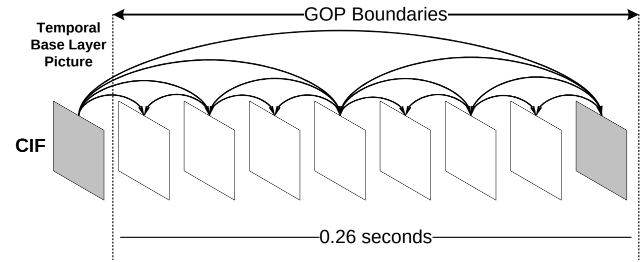

Fig. 8 Structure of a GOP of 8 in a CIF sequence. The pictures within the temporal base layer are represented in grey.

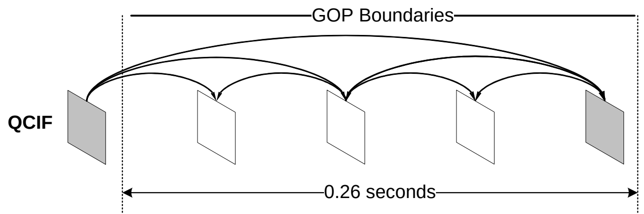

Fig. 9 Structure of a GOP of 4 in a QCIF sequence. The pictures within the temporal base layer are represented in grey.

This proposal is applied only to the temporal layers that satisfy the following condition:

$$n = \log_2(GOP_{size}) - k, with\, n > 0\, and\, k \in \{0,1\}$$

where n is the identifier of the temporal layer and k varies between 0 and 1. The remaining temporal layers will be decoded and re-encoded completely. In general, these layers are the two enhancement layers with the highest identifiers. This is due to the fact that most of the SVC encoding time is spent on the higher temporal enhancement layers as is shown in Figure 11 to Figure 14.

Moreover, a trade-off between time saving, bitrate increase and loss of PSNR is achieved when our approach is applied in these temporal layers as is shown in Figure 15 to Figure 18 for Baseline Profile and from Figure 23 to Figure 26 for Main Profile. To obtain the impact of the number of temporal layers to be transcoded in our proposal, the present approach was applied on different combinations of temporal layers while the remaining layers were decoded and re-encoded completely. A fixed GOP size was chosen, GOP = 8 for QCIF resolution and 16 for CIF, so QCIF sequences were composed of four temporal layers and CIF sequences of five. This GOP selection corresponds to having a picture of the temporal base layer roughly every 0.5s.

Tables 1-10 show the ΔPSNR, ΔBitrate and Time saving results when our technique is applied using different GOP sizes compared to the more complex reference transcoder in Baseline Profile. Tables 11-20 show the same, but for Main Profile. Time Savings are calculated for the full sequence and for the temporal layers where the approach is applied (the two last enhancement temporal layers as mentioned previously). ΔPSNR and ΔBitrate are calculated as specified in [20]. For PSNR, the averaged PSNR values of luminance (Y) and chrominance (U, V) are used. This global average PSNR is based on:

$$PSNR = \frac{4 \cdot PSNR_Y + PSNR_U + PSNR_V}{6}$$

To evaluate the time saving of the proposal, the following equation is calculated where Tref denotes the coding time used by the SVC reference software encoder and Tpro is the time spent by the proposed algorithm. The total time saving is calculated over the entire sequence, whereas for partial time saving only the time spent in the temporal layers where the proposal is applied (the two last enhancement temporal layers as mentioned previously) is taken into account. The equation applied is the following:

$$Time\,Saving(\%) = \frac{T_{pro} - T_{ref}}{T_{ref}} \cdot 100$$

All the results of Tables 1-10 for Baseline Profile are also collected from Figure 19 to Figure 22 where the increment of bitrate, loss of PSNR and Time Saving (partial and total) are represented with respect to the GOP size. The same procedure was followed to present the results of Tables 11-20 for Main Profile from Figure 27 to Figure 30 .

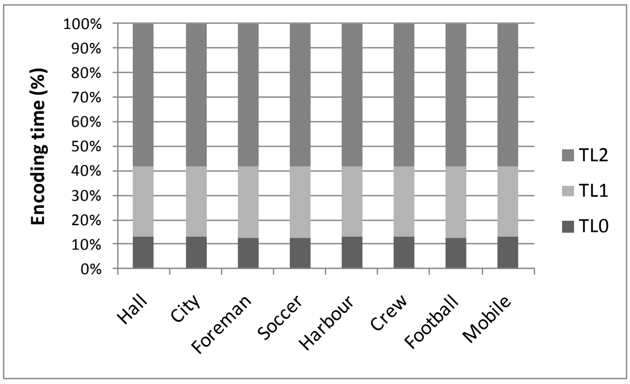

Fig. 10 Encoding Time for each Temporal Layer (TL) in % with QCIF resolution, GOP = 4 and Baseline Profile.

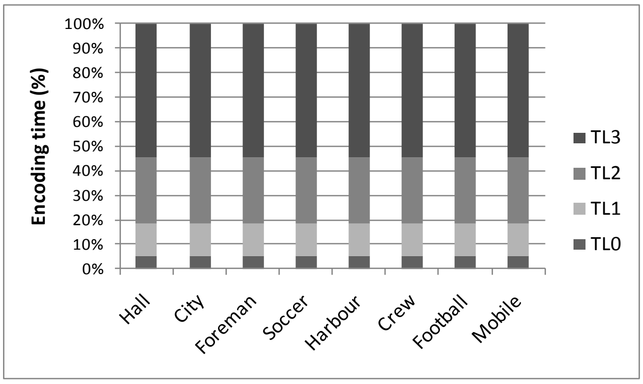

Fig. 11 Encoding Time for each Temporal Layer (TL) in % with QCIF resolution, GOP = 8 and Baseline Profile.

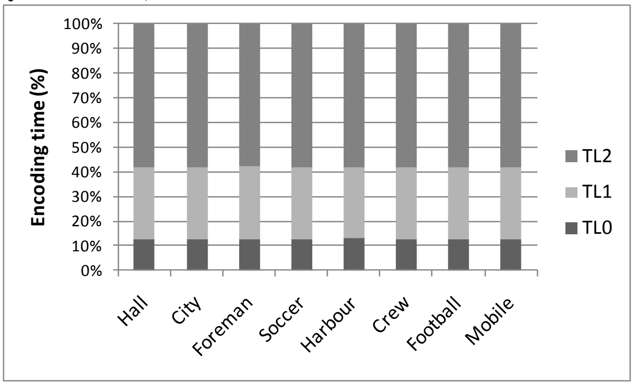

Fig. 12 Encoding Time for each Temporal Layer (TL) in % with CIF resolution, GOP = 4 and Main Profile.

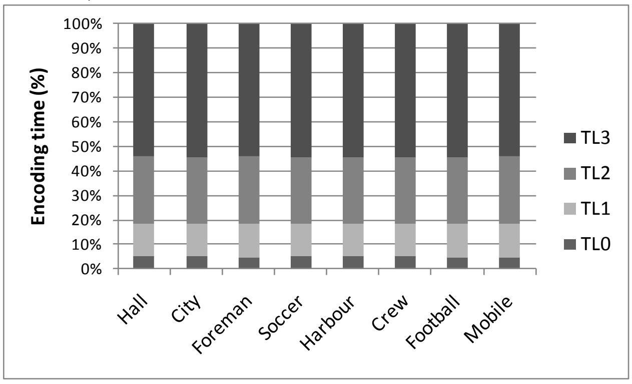

Fig. 13 Encoding Time for each Temporal Layer (TL) in % with CIF resolution, GOP = 8 and Main Profile.

As shown in those figures, for Baseline Profile the higher average PNSR lost over the reference is 0.05 dB with an average of bitrate around 1.5% in QCIF and 1.97% in CIF resolution achieving around 49% of reduction of computational complexity in the full sequence and 63% in the specific layers where the approach is applied. For Main Profile, the higher average PSNR lost over the reference is 0.05 dB, with an average increase of bitrate around 1.4% in QCIF and 2.2% in CIF resolution and achieving around 47% of reduction of computational complexity in the full sequence and 60% in the specific layers where the approach is applied. Moreover, we can conclude that for both profiles the impact of the GOP size in the global results is negligible.

6 Conclusions

In this paper, a technique for transcoding from H.264/AVC bitstream to an SVC bitstream with temporal scalability is proposed. This approach is based on accelerating the motion estimation in the SVC encoding process by reusing information available after decoding the H.264/AVC bitstream and is applied to a maximum of two enhancement temporal layers.

Tests using different CIF and QCIF sequences with varying GOP sizes have been run in Baseline and Main Profile. The obtained results show that a time saving around 60% is achieved independently of the GOP size for Baseline profile and 63% for Main Profile while coding efficiency is maintained in both cases. Moreover, this technique can be used to transcode from an H.264/AVC bitstream without temporal scalability to an H.264/AVC bitstream with temporal scalability.

Acknowledgements This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under grants CSD2006-00046, TIN2009-14475-C04, and it was also partly supported by JCCM funds under grants PEII09-0037-2328 and PII2I09-0045-9916.

References

1. M. Lindeberg, S. Kristiansen, T. Plagemann, and V. Goebel, “Challenges and techniques for video streaming over mobile ad hoc networks”. Multimedia Systems, vol. 17, n.1, pp. 51 --82, 2010.
2. L. Herranz, “Integrating semantic analysis and scalable video coding for efficient content-based adaptation”. Multimedia Systems, vol. 13 n.2, pp. 103--118, 2007.
3. D. De Schrijver, C. Poppe, S. Lerouge, W. De Neve and R. Van de Walle, “MPEG-21 bitstream syntax descriptions for scalable video codecs”, Multimedia Systems, vol. 11, n.5, pp. 403--421, 2006.
4. ITU-T and ISO/IEC JTC 1: Advanced Video Coding for Generic Audiovisual Services. ITU-T Rec. H.264/AVC and ISO/IEC 14496-10 (including SVC extension). March 2009.
5. A. Vetro, C. Christopoulos, H. Sun, “Video Transcoding Architectures and Techniques: an Overview”. IEEE Signal Processing, vol. 20 n.2, pp 18--29, 2003.
6. J. De Cock, S. Notebaert, K. Vermeirsch, P. Lambert, R. Van de Walle, “Dyadic spatial resolution reduction transcoding for H.264/AVC”. Multimedia Systems, vol.16 n.2, pp. 139 --149, 2010.
7. J. De Cock, S. Beheydt, J. De Lameillieure, R. Van de Walle, “Mixed architectures for H.264/AVC digital video transrating”. Multimedia Tools and Applications, vol. 44 n.1, pp. 39--64, 2009.
8. J. Bialkowski, M. Barkowsky, A. Kaup, “Fast video transcoding from H.263 to H.264/MPEG-4 AVC”. Multimedia Tools and Applications, vol. 35 n.2, pp. 127--146, 2007.
9. J. Xin, J. Li, A. Vetro, S. Sekiguchi, “Motion mapping and mode decision for MPEG-2 to H.264/AVC transcoding”. Multimedia Tools and Applications, vol. 35 n.2, pp. 203 -- 223, 2007.

10. H. Shen, S. Xiaoyan, F. Wu, H. Li, and S. Li, “Transcoding to FGS Streams from H.264/AVC Hierarchical B-Pictures”. IEEE Int. Conf. Image Processing, Atlanta, 2006.
11. J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, “Architectures of Fast Transcoding of H.264/AVC to Quality-Scalable SVC Streams”. IEEE Transactions on Multimedia vol. 11 n.7, pp.1209--1224, 2009.
12. R. Sachdeva, S. Johar, and E. Piccinelli, “Adding SVC Spatial Scalability to Existing H.264/AVC Video”. 8th IEEE/ACIS International Conference on Computer and Information Science, Shanghai, 2009.
13. A. Dziri, A. Diallo, M. Kieffer, and P. Duhamel, “P-Picture Based H.264 AVC to H.264 SVC Temporal Transcoding”. International Wireless Communications and Mobile Computing Conference, 2008.
14. H. Al-Muscati, and F. Labeau, “Temporal Transcoding of H.264/AVC Video to the Scalable Format”. 2nd Int. Conf. on Image Processing Theory Tools and Applications, Paris, 2010.
15. R. Garrido-Cantos, J. De Cock, J.L Martinez, S. Van Leuven, P. Cuenca, A. Garrido, R. Van de Walle, “On the Impact of the GOP size in an H.264/AVC-to-SVC Transcoder with Temporal Scalability”. Int. Conf. on Advances in Mobile Computing & Multimedia, Paris, 2010.
16. H. Schwarz, D. Marpe, T. Wiegand, “Overview of the Scalable Extension of the H.264/MPEG-4 AVC Video Coding Standard”. Joint Video Team, Doc. JVT-W132, San Jose, CA, USA, 2007.
17. H. Schwarz, D. Marpe, and T. Wiegand, Analysis of Hierarchical B pictures and MCTF. IEEE Int. Conf. Multimedia and Expo, Toronto, 2006.
18. Joint Video Team JSVM reference software, http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm
19. Joint Model JM reference software, http://iphome.hhi.de/suehring/tml/download/
20. G. Sullivan and G. Bjøntegaard, “Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material”. ITU-T VCEG, Doc. VCEG-N81. September 2001

Baseline Profile

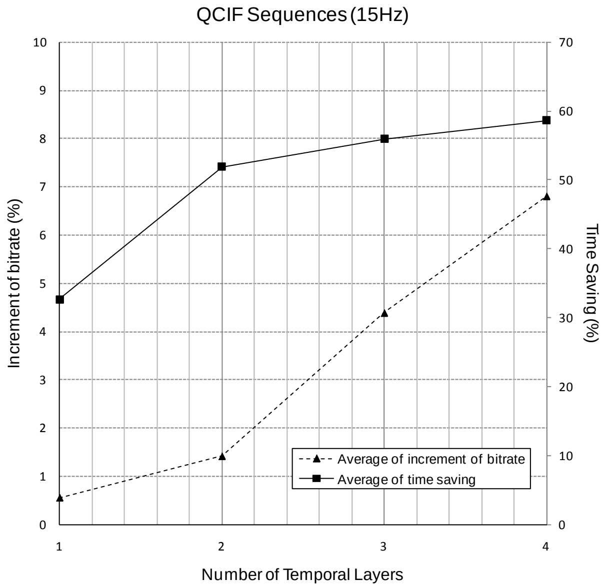

Fig. 14 Average of increment of bitrate and time saving depending on the number of layers transcoded for QCIF sequences and GOP = 8

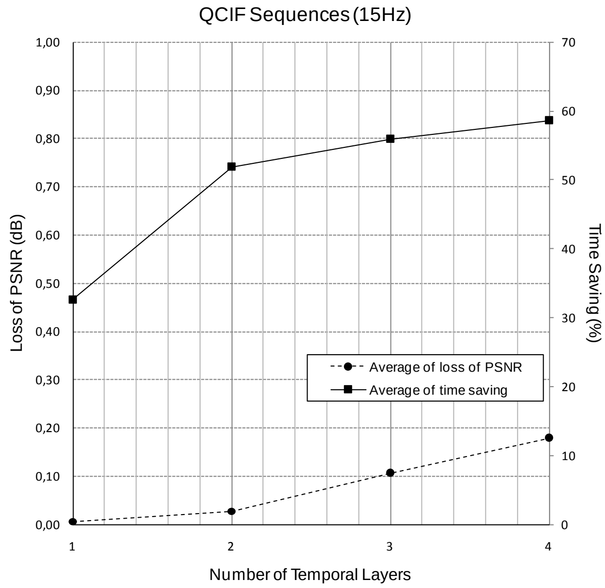

Fig. 15 Average of loss of PSNR and time saving depending on the number of layers transcoded for QCIF sequences and GOP = 8

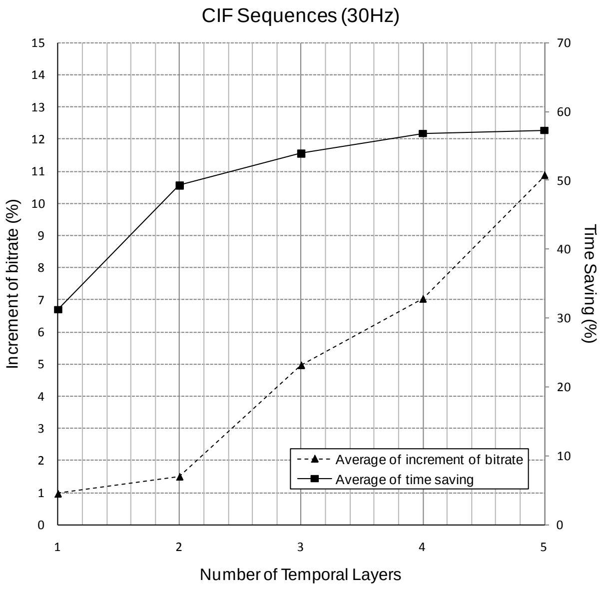

Fig. 16 Average of increment of bitrate and time saving depending on the number of layers transcoded for CIF sequences and GOP = 16

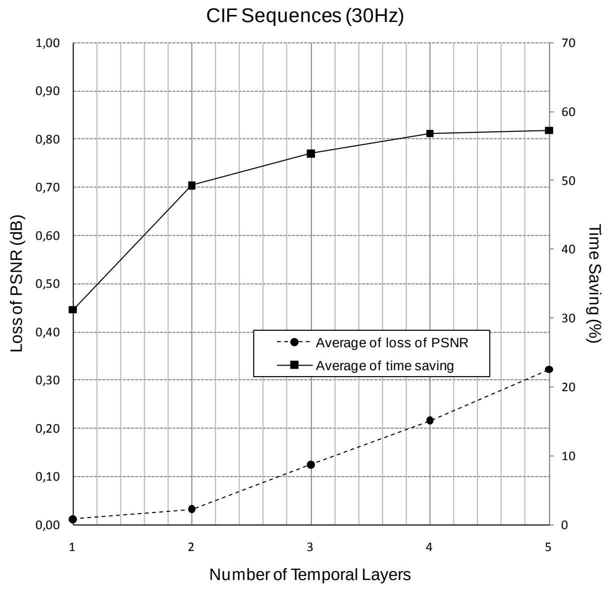

Fig. 17 Average of loss of PSNR and time saving depending on the number of layers transcoded for CIF sequences and GOP = 16

Table 1 RD performance and time saving of the improved transcoder with GOP = 2 and QCIF resolution

RD performance - GOP = 2 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.004	0.16	61.50	91.04
City	-0.006	0.62	55.35	82.01
Foreman	-0.009	0.37	42.92	63.71
Soccer	-0.079	2.92	31.26	46.41
Harbour	0.007	-0.04	60.68	89.97
Crew	-0.017	0.69	33.97	49.98
Football	-0.021	0.79	17.76	26.59
Mobile	0.003	-0.09	58.83	87.13
Average	-0.016	0.68	45.28	67.11

Table 2 RD performance and time saving of the improved transcoder with GOP = 4 and QCIF resolution

RD performance - GOP = 4 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.001	0.40	75.64	87.41
City	-0.055	1.66	67.27	77.73
Foreman	-0.006	0.81	50.72	58.76
Soccer	-0.093	3.94	36.56	42.43
Harbour	0.008	0.13	74.67	86.28
Crew	-0.043	1.71	39.03	45.41
Football	-0.031	1.15	19.32	22.60
Mobile	-0.023	0.99	71.72	82.79
Average	-0.031	1.35	54.37	62.93

Table 3 RD performance and time saving of the improved transcoder with GOP = 8 and QCIF resolution

RD performance - GOP = 8 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.011	0.59	70.51	87.46
City	-0.075	2.06	78.50	78.50
Foreman	0.015	0.75	45.19	56.27
Soccer	-0.076	4.48	33.40	41.92
Harbour	0.005	-0.18	69.61	86.29
Crew	-0.022	1.48	35.41	44.49
Football	-0.030	1.24	16.05	20.80
Mobile	-0.020	0.90	66.68	82.77
Average	-0.027	1.42	51.92	62.31

Table 4 RD performance and time saving of the improved transcoder with GOP = 16 and QCIF resolution

RD performance - GOP = 16 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.012	0.51	67.48	86.62
City	-0.167	3.21	60.59	77.79
Foreman	0.023	0.89	44.76	57.66
Soccer	-0.107	5.02	32.65	42.30
Harbour	0.040	0.22	66.69	85.51
Crew	-0.053	2.30	35.49	45.98
Football	-0.028	1.31	17.88	22.41
Mobile	-0.029	1.40	63.92	82.00
Average	-0.042	1.86	48.68	62.53

Table 5 RD performance and time saving of the improved transcoder with GOP = 32 and QCIF resolution

RD performance - GOP = 32 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.004	0.54	66.85	87.34
City	-0.111	2.41	60.84	79.25
Foreman	-0.035	0.84	41.21	53.96
Soccer	-0.136	5.87	35.11	45.72
Harbour	0.047	0.27	66.38	86.30
Crew	-0.081	3.22	36.73	48.29
Football	-0.028	1.35	16.36	22.18
Mobile	-0.095	2.74	63.60	82.95
Average	-0.054	2.16	48.39	63.25

Table 6 RD performance and time saving of the improved transcoder with GOP = 4 and CIF resolution

RD performance - GOP = 4 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.000	0.19	77.49	89.43
City	-0.068	3.33	63.57	73.38
Foreman	-0.026	1.19	51.51	59.54
Soccer	-0.126	4.92	38.88	44.99
Harbour	0.020	0.01	75.04	86.54
Crew	-0.043	1.73	38.33	45.87
Football	-0.040	1.59	20.68	24.20
Mobile	-0.018	0.89	69.77	80.45
Average	-0.038	1.73	54.41	63.05

Table 7 RD performance and time saving of the improved transcoder with GOP = 8 and CIF resolution

RD performance - GOP = 8 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.003	0.26	68.39	85.57
City	-0.051	3.61	58.61	66.91
Foreman	-0.056	1.48	43.22	53.93
Soccer	-0.105	4.97	36.10	42.45
Harbour	-0.005	0.25	69.66	86.47
Crew	-0.043	1.81	32.50	41.23
Football	-0.035	1.81	19.41	24.55
Mobile	-0.018	0.98	62.73	82.90
Average	-0.040	1.90	48.83	60.50

Table 8 RD performance and time saving of the improved transcoder with GOP = 16 and CIF resolution

RD performance - GOP = 16 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.066	0.42	66.51	86.09
City	-0.181	3.28	56.98	73.81
Foreman	-0.049	1.20	44.89	58.53
Soccer	-0.101	4.76	33.49	43.88
Harbour	0.135	-3.41	66.19	86.12
Crew	-0.060	2.01	43.01	49.24
Football	-0.041	2.28	21.32	24.48
Mobile	-0.021	1.56	61.73	80.03
Average	-0.032	1.51	49.27	62.77

Table 9 RD performance and time saving of the improved transcoder with GOP = 32 and CIF resolution

RD performance - GOP = 32 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.256	0.51	65.43	86.09
City	-0.178	5.12	56.09	73.89
Foreman	-0.049	1.20	44.89	58.53
Soccer	-0.132	5.03	33.47	44.63
Harbour	0.059	0.21	65.40	86.11
Crew	-0.081	2.61	43.21	49.28
Football	-0.041	2.25	20.85	24.02
Mobile	-0.014	1.58	60.74	80.06
Average	-0.023	2.31	48.76	62.83

Table 10 RD performance and time saving of the improved transcoder with GOP = 64 and CIF resolution

RD performance - GOP = 64 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.000	0.64	65.29	86.53
City	-0.072	2.93	56.69	75.24
Foreman	-0.034	1.16	41.27	55.16
Soccer	-0.198	7.67	34.48	46.31
Harbour	0.047	0.30	65.22	86.50
Crew	-0.071	2.89	43.08	49.34
Football	-0.040	1.98	20.83	24.14
Mobile	0.030	1.45	60.85	80.81
Average	-0.042	2.38	48.46	63.00

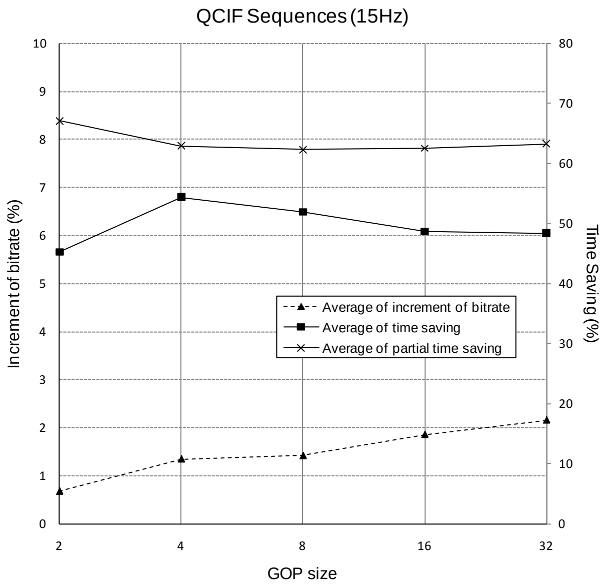

Fig. 18 Increment of bitrate and time saving depending on the GOP size (QCIF sequences)

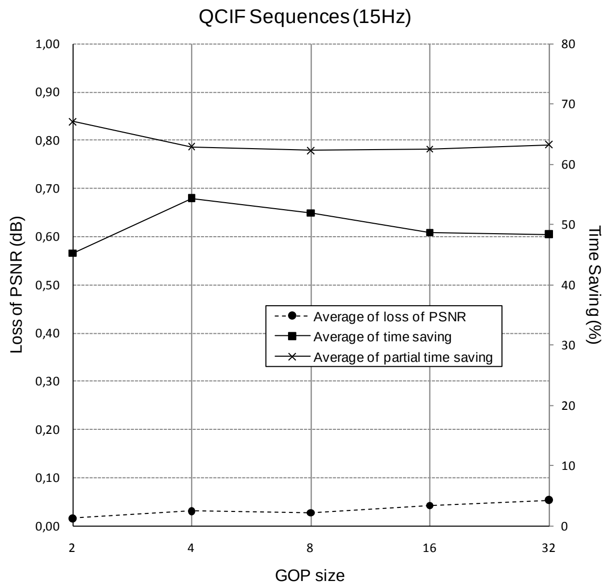

Fig. 19 Loss of PSNR and time saving depending on the GOP size (QCIF sequences)

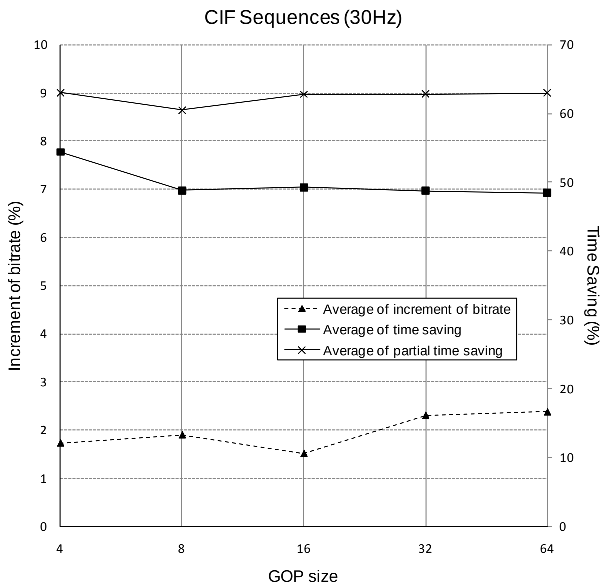

Fig. 20 Increment of bitrate and time saving depending on the GOP size (CIF sequences)

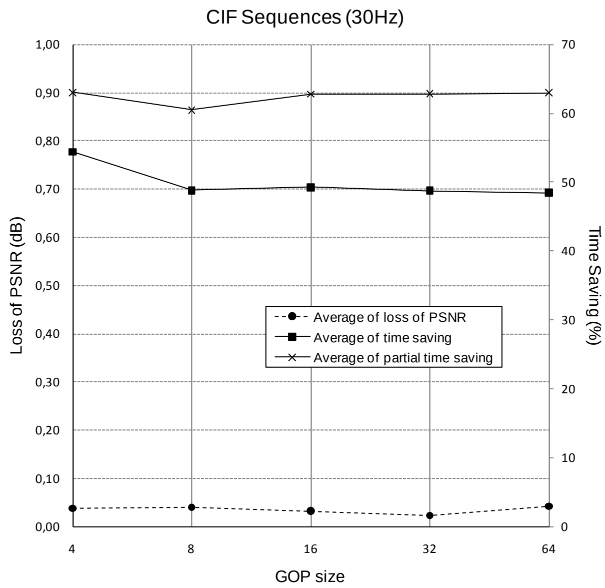

Fig. 21 Loss of PSNR and time saving depending on the GOP size transcoded (CIF sequences)

Main Profile

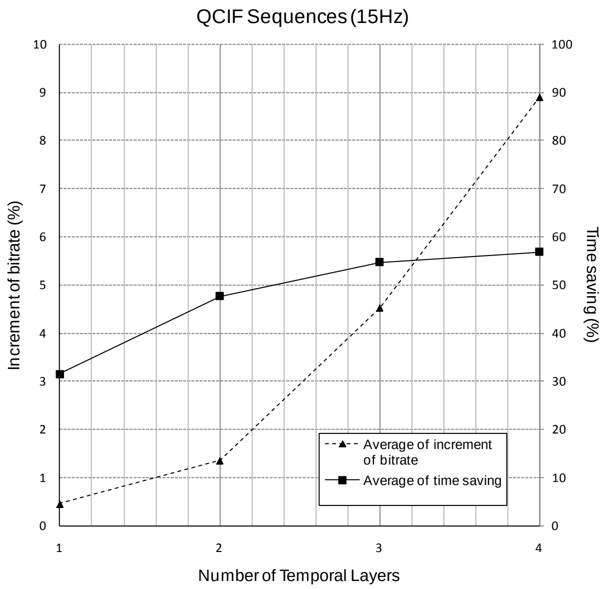

Fig. 22 Average of increment of bitrate and time saving depending on the number of layers transcoded for QCIF sequences and GOP = 8

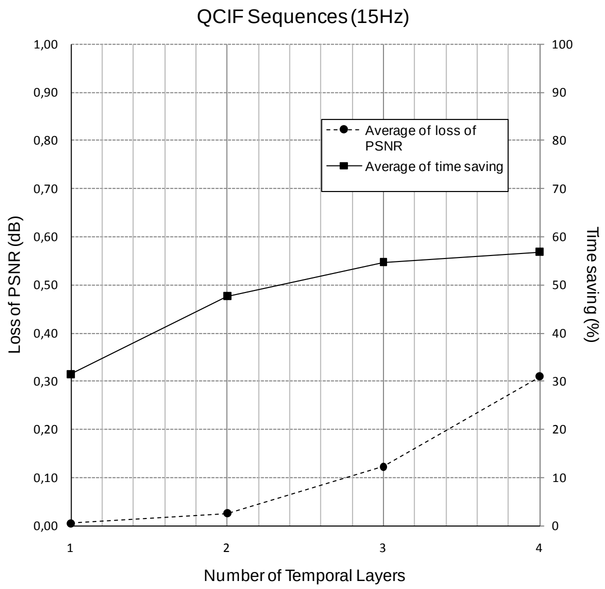

Fig. 23 Average of loss of PSNR and time saving depending on the number of layers transcoded for QCIF sequences and GOP = 8

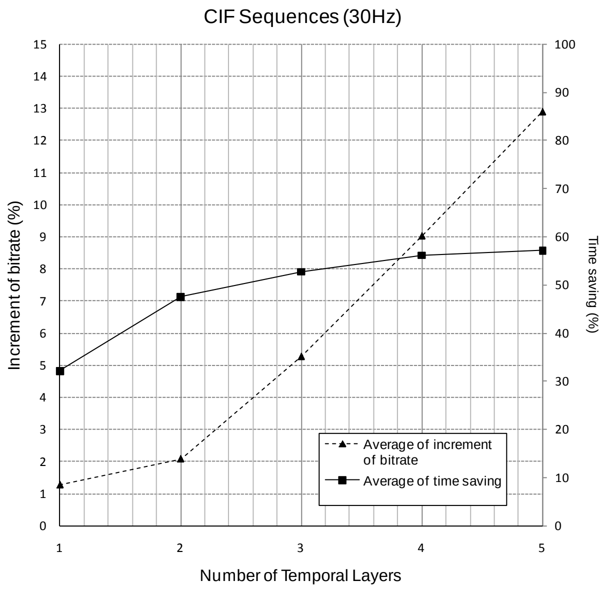

Fig. 24 Average of increment of bitrate and time saving depending on the number of layers transcoded for CIF sequences and GOP = 16

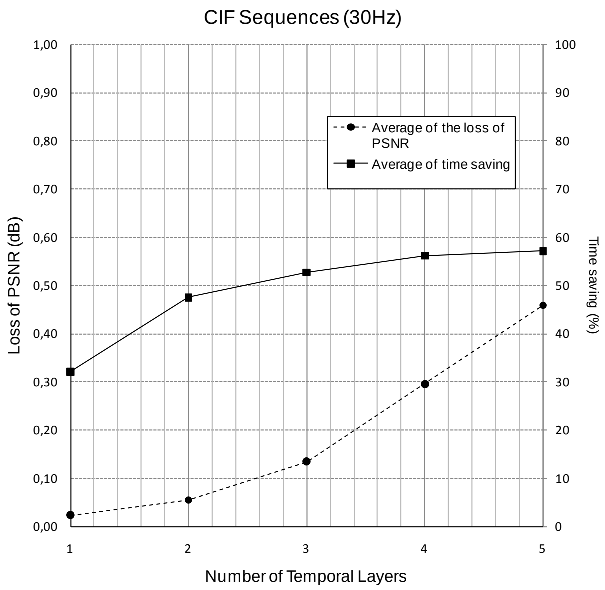

Fig. 25 Average of loss of PSNR and time saving depending on the number of layers transcoded for CIF sequences and GOP = 16

Table 11 RD performance and time saving of the improved transcoder with GOP = 2 and QCIF resolution

RD performance - GOP = 2 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.007	0.10	64.00	90.38
City	0.004	0.28	45.18	65.34
Foreman	-0.008	0.37	41.83	58.62
Soccer	-0.061	2.43	29.67	42.19
Harbour	0.003	0.05	59.89	86.96
Crew	-0.015	0.67	26.58	38.01
Football	-0.031	1.00	20.20	28.46
Mobile	0.000	0.03	62.67	88.14
Average	-0.013	0.62	43.75	62.26

Table 12 RD performance and time saving of the improved transcoder with GOP = 4 and QCIF resolution

RD performance - GOP = 4 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.016	0.60	77.64	88.33
City	-0.039	1.38	55.08	63.40
Foreman	-0.028	1.08	48.07	55.26
Soccer	-0.111	4.23	35.12	40.41
Harbour	0.003	0.32	74.94	86.24
Crew	-0.041	1.71	30.36	35.02
Football	-0.031	1.19	20.48	23.65
Mobile	-0.022	0.85	74.95	86.20
Average	-0.032	1.42	52.08	59.81

Table 13 RD performance and time saving of the improved transcoder with GOP = 8 and QCIF resolution

RD performance - GOP = 8 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.022	0.62	72.27	89.35
City	-0.028	1.66	46.74	60.40
Foreman	-0.010	0.92	41.83	51.83
Soccer	-0.123	4.13	34.19	41.39
Harbour	0.005	0.32	70.25	86.32
Crew	-0.026	1.29	27.09	33.77
Football	-0.029	1.17	19.08	23.71
Mobile	-0.018	0.76	69.68	86.13
Average	-0.026	1.36	47.64	59.11

Table 14 RD performance and time saving of the improved transcoder with GOP = 16 and QCIF resolution

RD performance - GOP = 16 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.013	0.59	69.56	88.58
City	-0.108	2.73	50.16	64.01
Foreman	-0.026	0.85	42.34	54.04
Soccer	-0.087	4.61	31.04	39.81
Harbour	0.023	0.33	67.03	85.45
Crew	-0.041	2.12	28.04	35.87
Football	-0.029	1.33	18.54	23.88
Mobile	-0.013	0.68	67.05	85.45
Average	-0.034	1.66	46.72	59.64

Table 15 RD performance and time saving of the improved transcoder with GOP = 32 and QCIF resolution

RD performance - GOP = 32 - QCIF (15 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.017	0.59	69.29	89.48
City	-0.030	0.92	52.04	67.27
Foreman	-0.010	0.93	37.39	48.45
Soccer	-0.105	5.74	32.63	42.30
Harbour	0.035	0.39	66.57	86.07
Crew	-0.061	3.19	29.13	37.79
Football	-0.027	1.36	18.39	24.09
Mobile	-0.009	0.70	67.09	86.57
Average	-0.028	1.73	46.57	60.25

Table 16 RD performance and time saving of the improved transcoder with GOP = 4 and CIF resolution

RD performance - GOP = 4 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.004	0.45	77.69	89.43
City	-0.126	3.26	59.61	67.90
Foreman	-0.041	1.47	53.34	60.72
Soccer	-0.135	5.64	36.81	42.65
Harbour	-0.003	0.28	74.80	86.69
Crew	-0.043	1.73	34.33	39.29
Football	-0.041	1.92	22.49	26.08
Mobile	-0.017	0.71	75.15	86.61
Average	-0.051	1.93	54.28	62.42

Table 17 RD performance and time saving of the improved transcoder with GOP = 8 and CIF resolution

RD performance - GOP = 8 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.007	0.48	68.38	85.58
City	-0.120	3.62	49.16	61.81
Foreman	-0.038	1.39	43.23	54.44
Soccer	-0.111	5.74	30.60	38.76
Harbour	0.012	0.26	66.43	83.21
Crew	-0.035	1.96	26.91	34.21
Football	-0.039	1.90	17.85	22.96
Mobile	-0.018	0.77	66.21	82.84
Average	-0.043	2.02	46.10	57.98

Table 18 RD performance and time saving of the improved transcoder with GOP = 16 and CIF resolution

RD performance - GOP = 16 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	0.003	0.64	66.01	85.43
City	-0.100	2.61	52.22	67.85
Foreman	-0.035	1.30	44.99	58.56
Soccer	-0.121	5.76	34.89	45.59
Harbour	0.011	0.31	64.40	83.43
Crew	-0.043	2.08	31.50	41.28
Football	-0.062	3.17	19.89	27.67
Mobile	-0.016	0.79	66.48	86.45
Average	-0.045	2.08	47.55	62.03

Table 19 RD performance and time saving of the improved transcoder with GOP = 32 and CIF resolution

RD performance - GOP = 32 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.002	0.75	64.79	85.26
City	-0.119	5.70	47.41	62.73
Foreman	-0.034	1.22	40.78	54.09
Soccer	-0.142	5.96	29.09	38.87
Harbour	0.016	0.40	62.90	82.83
Crew	-0.048	2.55	26.04	34.94
Football	-0.037	1.98	16.75	22.85
Mobile	-0.015	0.82	62.61	82.45
Average	-0.048	2.42	43.80	58.00

Table 20 RD performance and time saving of the improved transcoder with GOP = 64 and CIF resolution

RD performance - GOP = 64 - CIF (30 Hz)				
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time Saving (%)	
			Full Seq.	Partial
Hall	-0.014	0.92	64.67	85.71
City	-0.071	2.83	48.31	64.35
Foreman	-0.026	1.09	36.89	49.39
Soccer	-0.127	8.34	30.19	40.62
Harbour	0.040	0.49	62.74	83.21
Crew	-0.081	3.79	27.41	36.97
Football	-0.039	2.07	19.58	25.96
Mobile	0.001	0.61	66.10	86.65
Average	-0.040	2.52	44.49	59.11

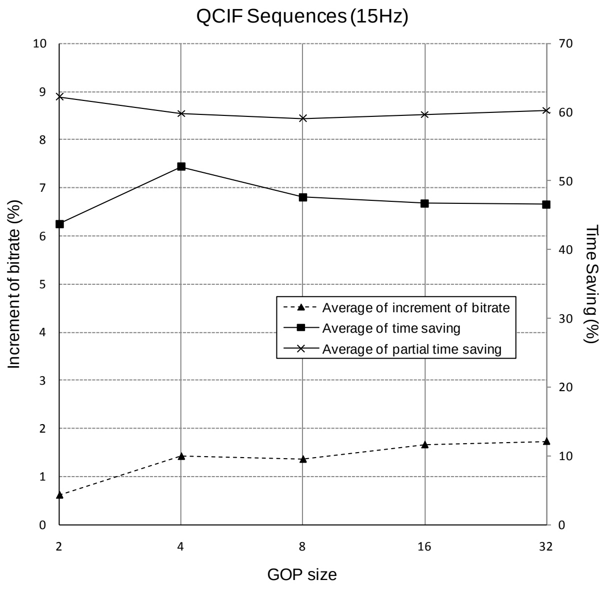

Fig. 26 Increment of bitrate and time saving depending on the GOP size (QCIF sequences)

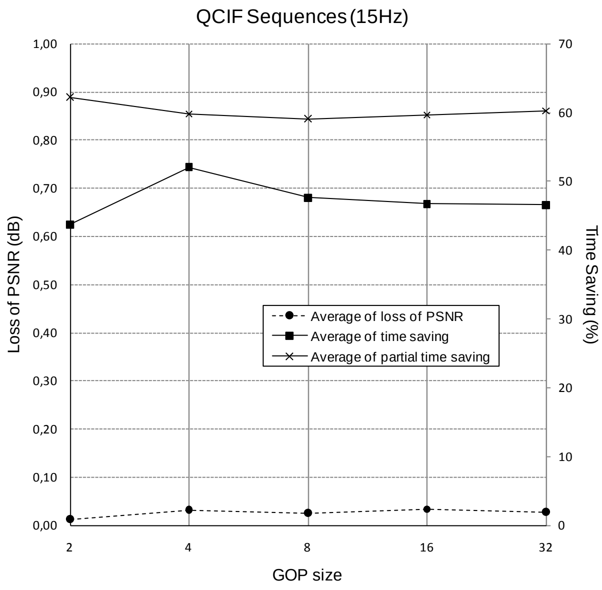

Fig. 27 Loss of PSNR and time saving depending on the GOP size (QCIF sequences)

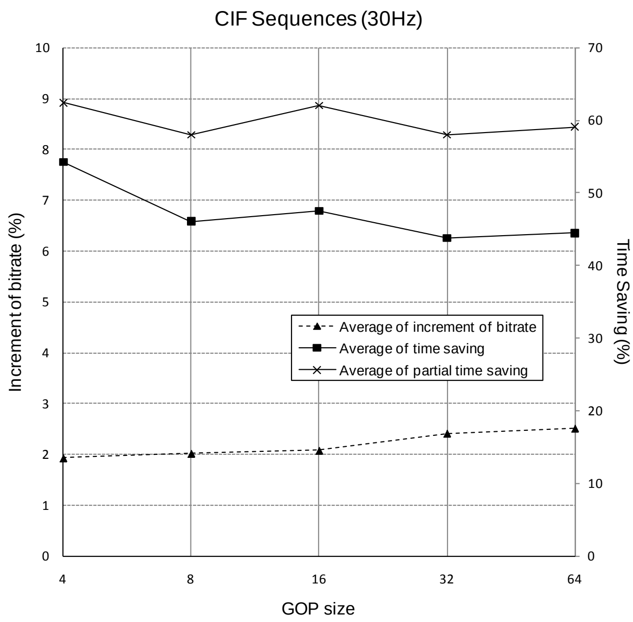

Fig. 28 Increment of bitrate and time saving depending on the GOP size (CIF sequences)

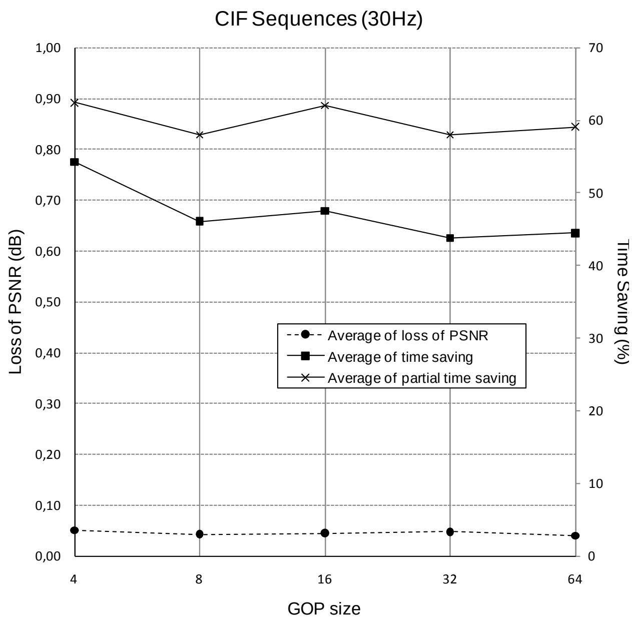

Fig. 29 Loss of PSNR and time saving depending on the GOP size transcoded (CIF sequences)