Temporal video transcoding from H.264/AVC-to-SVC for digital TV broadcasting

R. Garrido-Cantos, J. De Cock, J. L. Martinez, S. Van Leuven, P. Cuenca, and A. Garrido

In: Telecommunication Systems, 61 (1), 21-41, 2016.

Telecommun Syst (2016) 61:21–41
DOI 10.1007/s11235-014-9950-z

Temporal video transcoding from H.264/AVC-to-SVC for digital TV broadcasting

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

Published online: 6 January 2015

Abstract Mobile digital TV environments demand flexible video compression like scalable video coding (SVC) because of varying bandwidths and devices. Since existing infrastructures highly rely on H.264/AVC video compression, network providers could adapt the current H.264/AVC encoded video to SVC. This adaptation needs to be done efficiently to reduce processing power and operational cost. This paper proposes two techniques to convert an H.264/AVC bitstream in Baseline (P-pictures based) and Main Profile (B-pictures based) without scalability to a scalable bitstream with temporal scalability as part of a framework for low-complexity video adaptation for digital TV broadcasting. Our approaches are based on accelerating the interprediction, focusing on reducing the coding complexity of mode decision and motion estimation tasks of the encoder stage by using information available after the H.264/AVC decoding stage. The results show that when our techniques are applied, the complexity is reduced by 98 % while maintaining coding efficiency.

Keywords Mobile digital TV · Videoadaptation · Data mining · H.264/AVC · Scalable video coding (SVC) · Temporal scalability

R. Garrido-Cantos (✉) · J. L. Martinez · 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

J. L. Martinez
e-mail: joseluismm@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
Multimedia Lab, Department of Electronics and Information Systems, Ghent University-iMinds,
Gaston Crommenlaan 8, Bus 201, Ledeberg,
9050 Ghent, Belgium
e-mail: jan.decock@ugent.be

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

1 Introduction

The users demand for multimedia content services has grown spectacularly in the last years. One of the most requested services is the digital TV (based on H.264/AVC) on mobile devices. There are different alternatives to transmit this content from broadcasters to the users. One of them is the mobile internet protocol television (IPTV) [23]. Other networks deployed specially for delivering multimedia contents to mobile terminals are advanced television systems committee-mobile/handheld (ATSC-M/H) [1] in North America and digital video broadcasting handheld (DVB-H) in Europe [8]. All of them are characterized by fluctuating bandwidths and varying device capabilities. Because of these irregularities, it is necessary to adapt the video stream to the changing environment [6,16,19].

Since this adaptability is not incorporated in H.264/AVC, scalable video coding (SVC) [15] was introduced. The SVC video stream is divided into layers, each adding more spatial, temporal or quality resolution. By removing layers, spatial resolution, frame rate, or quality can be reduced. On the one hand, this adaptation must occur in terms of bitrate to adapt to the constraints of the transmission due to the dynamic nature of the links of the network and on the other hand, in terms of bitrate or spatial resolution to fit into the different capabilities of a mobile terminal (battery lifetime, computing capacity, or screen resolutions). Therefore, real time video adap-tation for mobile devices plays a crucial role.

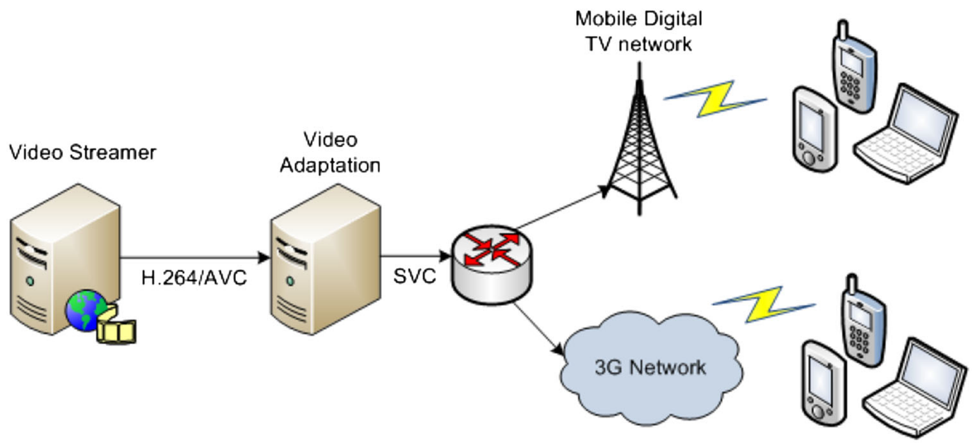


Fig. 1 Example of an SVC transcoder for digital video broadcasting

Nowadays, broadcasting for TV and mobile TV is largely based on H.264/AVC. As a result, to extend existing infrastructure with scalable capabilities, H.264/AVC-to-SVC transcoding is needed. Video transcoding [28] can be regarded as a process for ef-ficient adaptation of media content, in order to match the properties and constraints of transmission networks and terminal devices, by efficiently (re)using information from the incoming bitstream, while at the same time minimizing quality loss due to the adaptation. A video transcoder is composed of a decoding stage followed by an encoding stage. The simplest transcoder is constructed by connecting a decoder which decodes the input bitstream with an encoder which forms a new bitstream with different characteristics. When applying this transcoding step at the broadcaster's premises, existing H.264/AVC infrastructure can be maintained and low complexity adaptations can be made in the broadcast network, where needed. The resulting network architecture is illustrated in Fig. 1. In this figure, an H.264/AVC-to-SVC transcoder (video adaptation) is added on the transition from the video streamer to the mobile digital TV network. The goal is to perform the required adaptation process faster than the concatenation of decoder and encoder. Its efficiency is obtained by reusing as much information as possible from the original bitstream, such as mode decisions and motion information, to reduce the encoding SVC time, focusing on the mode decision process and the estimation process.

In particular, in this paper we propose a low complexity transcoder for transforming H.264/AVC bit-streams in Baseline (P-picture based) and Main Profile (B-picture based) without temporal scalability (frame rate variability) into an SVC bitstream with temporal scalability.

The low complexity transcoder is based on to accelerate the interprediction, focusing on reducing the coding complexity of mode decision and motion estimation tasks of the encoder stage by reusing information such as residual data, motion vectors, mode decision, etc. in the H.264/AVC decoding stage. For accelerating the mode decision task, a decision tree for every profile for narrowing down the MacroBlock (MB) types to be checked by the SVC encoder were developed. These decision trees have been obstained using machine learning (ML) tools. On the other hand, for accelerating the motion estimation task, a reduced area was created dynamically for every MB and sub-MB.

The remainder of this paper is organized as follows. In Sect. 2, the state-of-the-art for H.264/AVC-to-SVC transcoding is discussed. Section 3 describes the technique background and temporal scalability in SVC. In Sect. 4 our approaches are depicted. In Sect. 5 the implementation results are shown. Finally, in Sect. 6 conclusions are presented.

2 Related work

In the last few years, different techniques for tran-scoding from H.264/AVC-to-SVC have been proposed. Some of the proposals related to quality-SNR scalability are described in [5,25–27]. There are few related to spatial [20] and temporal scalability [2,7,9–11]. Since this proposal focuses on transcoding for providing temporal scalability, only techniques related to that type are explained with more detail in the following lines.

In 2008 a transcoding method from an H.264/AVC P-picture-based bitstream to an SVC bitstream was presented in [7] by Dziri et al. In this approach, the H.264/AVC bitstream was transcoded to two layers of P-pictures (one with reference pictures and the other with non-reference ones). Then, this bitstream was transformed to an SVC bitstream by syntax adaptation. In 2010, Al-Muscati et al. proposed another technique for transcoding that provided temporal scalability in [2]. The method presented was applied in the Baseline Profile and reused information from the mode decision and motion estimation processes from the H.264/AVC stream. During that year some we presented more proposals such as 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 the SVC encoding algorithm by reducing the motion estimation process time. The approach was developed for Main Profile and dynamically adapted for several temporal layers [9]. Later, in 2011, the previous algorithm was adjusted for the Baseline Profile and P frames [10]. The same year, a technique for accelerating the mode decision task of SVC encoder in an H.264/AVC-to-SVC transcoder with temporal scalability and Baseline Profile was presented in [11] and then, extended for working together with [10] in [12]. In 2012, Chia-Hung and al. proposed another technique [3] for transcoding from H.264/AVC-to-SVC using probability models and Markov chains and we presented another work adapting [13] for working in Main Profile. The present work extends the evaluation made in [13] using more GOP sizes and combined with [12], presenting all the results for applying this technique to different sequences.

3 Technical background

3.1 Temporal scalability in scalable video coding

H.264/AVC was an enabling technology for digital video in almost every area that was not previously covered by H.262/MPEG-2, and has substantially displaced the older standard within its existing application domain. It is widely used for many applications, including broadcasting of High Definition (HD) TV signals over satellite, cable, terrestrial transmission systems, video content acquisition and editing systems, camcorders, security applications, Internet and mobile network video, Blu-ray discs, and real-time conversational applications such as video chat, video conferencing, and telepresence systems. H.264/AVC employs a hybrid block-based video compression technique which is based on combining picture Inter prediction to exploit temporal redundancy and transform-based coding of the prediction errors to exploit spatial redundancy.

As said previously, SVC is an extension of H.264/AVC. SVC streams are composed of layers which can be removed to adapt the streams to the needs of end users or the capabilities of the terminals or the network conditions. The layers are divided in-to one base layer and one or more enhancement layers which employ data of lower layers for efficient coding. SVC supports three main types of scalability:

– *Temporal Scalability* The base layer is coded at a low frame rate. By adding enhancement layers the frame rate of the decoded sequence can be increased.
– *Spatial Scalability* The base layer is coded at a low spatial resolution. By adding enhancement layers the resolution of the decoded sequence can be increased.
– *Quality (SNR) Scalability* The base layer is coded at a low quality. By adding enhancement layers the quality of the decoded sequences can be increased.

To identify to which layer each frame is associated, a layer identifier triplet (D,T,Q) is transmitted for every frame. In this triplet, D represents the dependency layer or spatial layer identifier (Did), T is the temporal layer identifier (Tid) and Q is the quality layer identifier (Qid). Each enhancement layer is placed in a new network abstraction layer unit (NAL-unit). Depending on the available bit rate or the device capabilities, NAL units are either routed to the end-user or dropped in the (congested) wireless network. Even when all packets arrive, the end user device can decide not to decode some enhancement layer packets (e.g., in order to reduce energy consumption).

Since our proposal focus on temporal scalability, a brief explanation of this type of scalability is given in this section. In a sequence with temporal scalability, the base layer represents the lowest frame rate (with an identifier equal to 0). With one or more temporal enhancement layers (with identifiers that increase by 1 in every layer), a higher frame rate can be achieved. Figure 2 shows a sequence encoded as four temporal layers. The base layer (layer 0) consists of frames 0 and 8 and provides 1/8 of the original frame rate. Frame 4 lies within the first enhancement temporal layer and, decoded together with layer 0, produces 1/4 of the frame rate of the full sequence. Layer 2 consists of frames 2 and 6; together with layers 0 and 1 it provides a frame rate that is 1/2 of the frame rate of the whole sequence.

Temporal scalability can be achieved using P and B coding tools that are available in H.264/AVC and by extension in SVC. Flexible prediction tools make possible to mark any picture as reference picture, so that it can be 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 [15] 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 was mentioned previously, the temporal base layer represents the lowest frame rate that can be obtained. The frame rate can be increased by adding pictures of the enhancement layers.

Temporal scalability based on P pictures was introduced in [29]. This technique provides lower latency and is particularly useful for multimedia communications like mobile

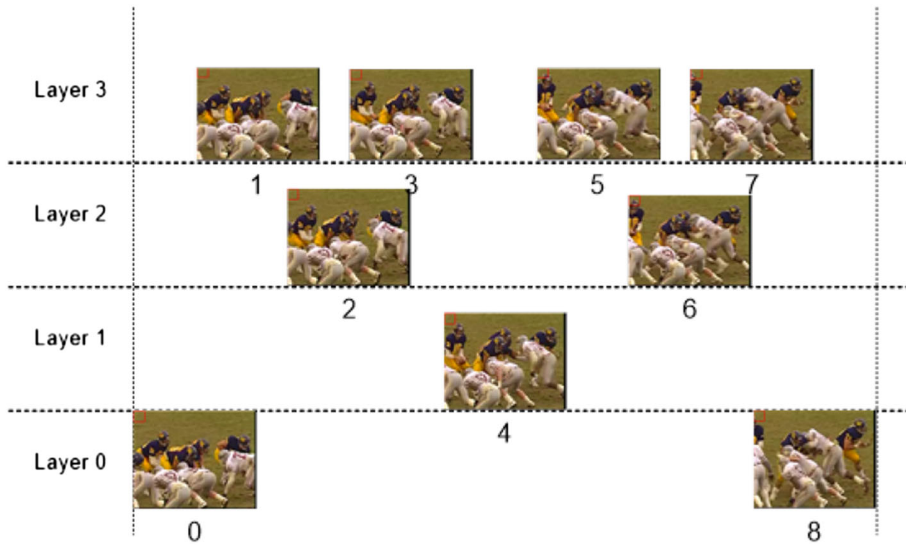


Fig. 2 Sequence with temporal scalability. Distribution of the eight first frames per every layer

video broadcasting or mobile digital television where the transmission of a scalable bitstream would be a good solution to address mobile terminals with several requirements. There are different structures for enabling temporal scalability, but the one used by default in the Joint Scalable Video Model (JSVM) reference encoder software [18] is based on hierarchical pictures with a dyadic structure where the number of temporal layers is thus equal to 1 + log2[GOP size]. For a comprehensive overview of the scalable extension of H.264/AVC, the reader is referred to [22].

3.2 Mode decision process

In H.264/AVC and its extension SVC, the pictures are partitioned into MBs. For every MB a prediction is created from previously encoded data and is subtracted from the MB to form a residual. By selecting the best prediction options for an individual MB, an encoder can minimize the residual size to produce a highly compressed bitstream.

H.264/AVC and SVC support both intra prediction and inter prediction. Intra prediction only requires data from the current picture, while inter prediction uses data from a picture that has previously been coded and transmitted (a reference picture) and is used for eliminating temporal redundancy in P and B frames.

SVC supports motion compensation block sizes ranging from 16×16, 16×8, 8×16 to 8×8; where each of the subdivided regions is an MB partition. If the 8×8 mode is chosen, each of the four 8×8 block partitions within the MB may be further split in 4 ways: 8×8, 8×4, 4×8 or 4×4, which are known as sub-MB partitions. Moreover, SVC also allows intra predicted modes, and a skipped mode in inter frames for referring to the 16×16 mode where no motion and residual information is encoded. Therefore, both H.264/AVC and SVC allow not only the use of the MBs in which the images are decomposed but also allow the use of smaller partitions by dividing the MBs in different ways. MB and sub-MB partitions for inter prediction are shown in Fig. 3.

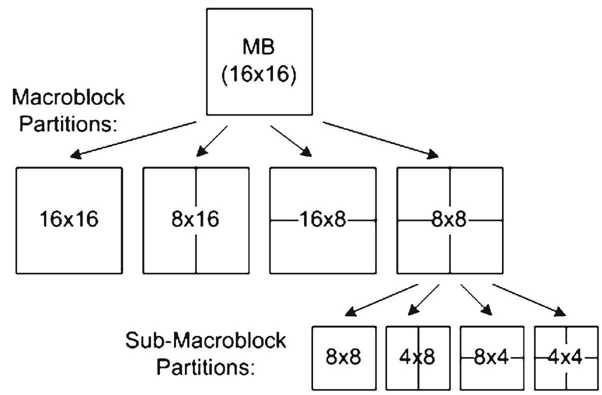


Fig. 3 Macroblock and sub-macroblock partitions for inter prediction

3.3 Motion estimation process

The motion estimation process consists in finding a region in a reference frame that matches the current MB as far as possible. In order to find this region, a search area situated in the reference frame is def-ned. That search area is centered on the current MB partition position, and the region within the search area that minimizes a matching criterion is chosen. For elimination of the temporal redundancy, motion vectors (MVs) between every MB or sub-MB partition and that block which generates the most appropriate match inside the search area of the reference frame are calculated. For each interpredicted sub-macroblock partition, one or two motion vectors can be provided, which indicates the prediction with quarter pixel accuracy. Thus, for any macroblock, one to six-

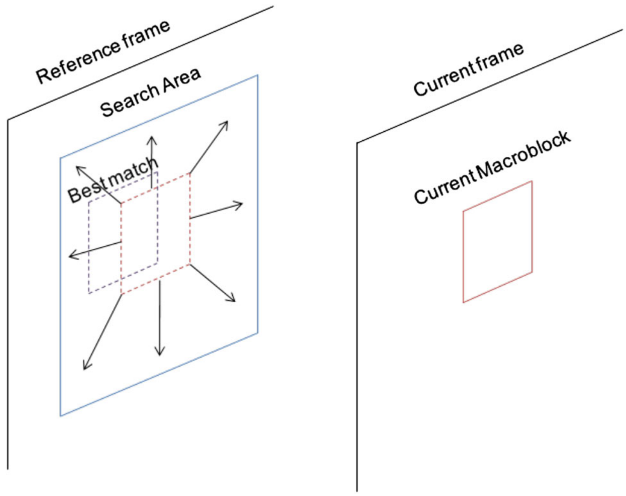


Fig. 4 Motion estimation process

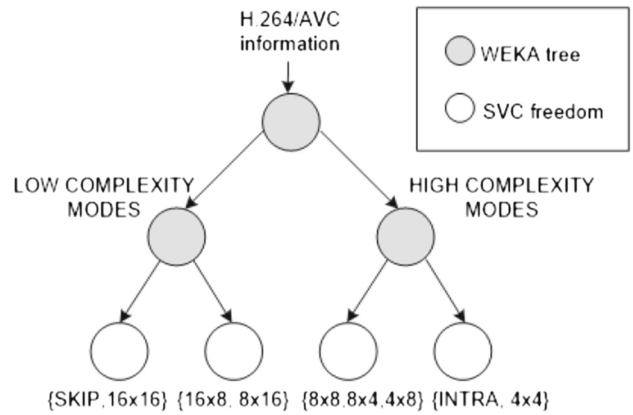


Fig. 5 Decision tree

teen motion vectors can be provided. Since for each sub-macroblock partition a different motion vector can be determined, a search operation is performed for each of these sub-macroblock partitions, resulting in a very complex operation. Reducing the motion vector search complexity is one of the means to reduce the complexity. The mode decision process evaluates the optimal motion vector for each sub-partition size, while all reference lists are evaluated (i.e, the mode decision process invokes the motion estimation process multiple times). The mode and reference list yielding the lowest RD is selected and the corresponding macroblock type is used to signal the syntactical information (such as macroblock type and MV) and residual data. Thus, all possible partitions and reference lists are evaluated, resulting in a very complex operation. The motion estimation process is illustrated in Fig. 4.

4 Proposed video transcoding

One of the key points that need to be addressed in the design of an efficient H.264/AVC-to-SVC transcoder is interprediction, since it is one of the most computationally-intensive tasks involved in the transcoding process. The interprediction includes mode decision and motion estimation processes described in Sect. 3.

The idea behind the proposed transcoder is to accelerate the interprediction, focusing on reducing the coding complexity of mode decision and motion estimation tasks of the encoder stage by using information available after the H.264/AVC decoding stage.

For accelerating the mode decision task, a decision tree for narrowing down the MB types to be checked by the SVC encoder was developed. This decision tree has been obtained using ML tools. On the other hand, for accelerating the motion estimation task, a reduced area was created dynamically for every MB and sub-MB. In the next subsections we will describe these two techniques.

4.1 Reducing mode decision complexity

Machine learning is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data. It has the decision making ability with low computation complexity, basically, if-then-else operations.

In this framework, we used ML tools in order to convert into rules the relationships between some data extracted from H.264/AVC decoding process and the MB mode partitioning of SVC (this could be seen as the variable to understand). By using these rules instead of the MB partition algorithm of the SVC encoder, we can speed up this process. In this paper, two different decision trees (one for Baseline Profile and another one for Main Profile) with three levels of decision are presented. These decision trees narrow the mode decisions that can be chosen by the standard.

To build every decision tree we used the WEKA software [14]. For every macroblock, the extracted information is used to generate the decision tree (and then to decide the macroblock partitioning). Some operations and statistics are calculated for this data:

- Residual of the whole macroblock.
- Length of the average of the motion vectors of a macroblock.
- Mean of variances of the residual of 4×4 blocks within a macroblock.
- Variance of means of the residual of 4×4 blocks within a macroblock.

The information enumerated above together with the SVC encoder mode decision was introduced and then, an ML classifier was run. In this case, the well-known RIPPER algorithm [4] was used. The training file was generated using the

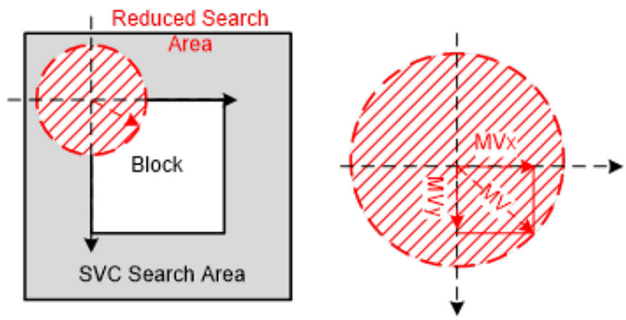


Fig. 6 Proposed reduced search area

sequence Football and only taking into account the frames within the two enhancement temporal layers with the highest identifiers. The binary decision tree obtained has three decision levels:

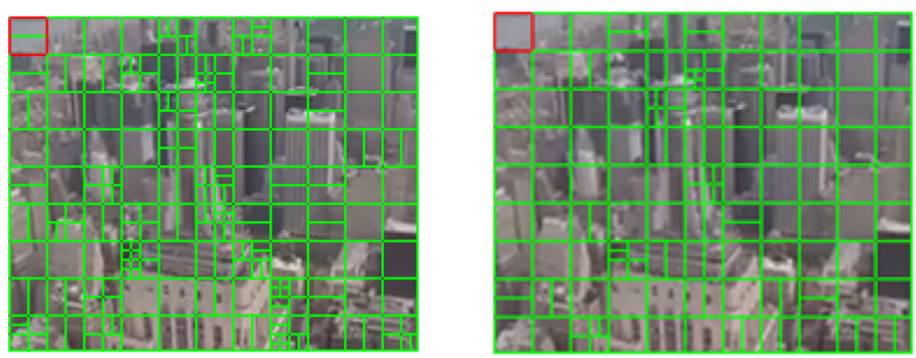
1. 1st level: Discriminates between LOW SKIP, 16 × 16, 16 × 8, 8 × 16 and HIGH COMPLEXITY INTRA, 8 × 8, 8 × 4, 4 × 8, 4 × 4 modes.
2. 2nd level: Inside the LOW COMPLEXITY bin, a decision between SKIP, 16 × 16 or 16 × 8, 8 × 16 is made.
3. 3rd level: Inside the HIGH COMPLEXITY bin, a decision between 8 × 8,8 × 4,4 × 8 or 4 × 4, INTRA is made.

This tree was generated with the information available after the decoding process and does not focus the final MB partition, but reduces the set of final MB that can be chosen by SVC encoder. This is represented in Fig. 5 where the white circles represent the set of MB partition where the reference standard can choose into. The process for generating both decision trees is the same as described above.

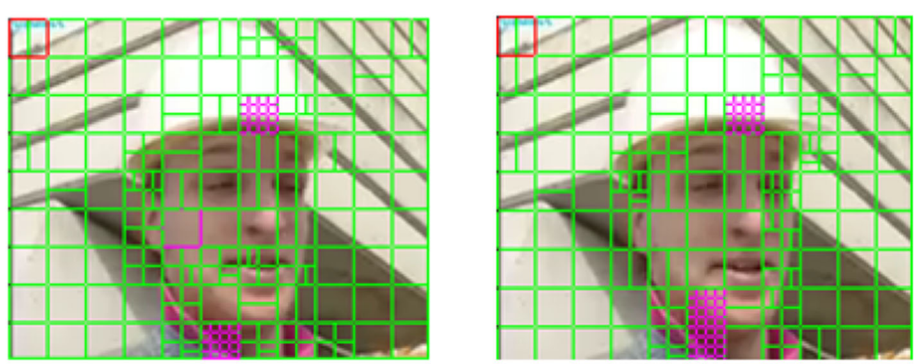
The ML process gave us a decision tree for Base-line Profile that classified correctly in about 91 % of cases in the 1st level, 87 % in the 2nd level and 80 % in the 3rd level and in about 93 % of cases in the 1st level, 89 % in the 2nd level and 84 % in the 3rd level for Main Profile.

These decision trees are composed of a set of thresholds for the H.264/AVC residual and for the statistics related to it. Since the MB mode decision, and hence the thresholds, depend on the Quantization Parameter (QP) used in the H.264/AVC stage, the residual, the mean and the variance threshold will be different at each QP. The solution is to develop a single decision tree for a QP and adjust the mean and the variance threshold used by the trees basing on the QP. Since the relationship between the quantization step size and the QP is well known, an adjustment to the decision tree can be made. The proposed transcoder uses a single decision tree developed for a mid-QP of 28, and which is later adjusted for other QPs (32, 36 and 40). Since the quantization step size

Fig. 7 Examples of MB partitions generated by H.264/AVC (*left*) and SVC (*right*)



(a) 4th frame of City QCIF sequence encoded in Main Profile



(b) 2nd frame of Foreman QCIF sequence encoded in Baseline Profile

Table 1 Encoding time (%) for each temporal layer (TL) with different GOP sizes using QCIF Baseline Profile

Encoding time (%) of every temporal layer QCIF (15 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	TL0	TL1	TL2	TL0	TL1	TL2	TL3	TL0	TL1	TL2	TL3	TL4
Hall	12.93	28.89	58.18	5.12	13.5	27	54.38	1.77	6.36	13.7	26.14	52.65
City	12.96	28.89	58.16	5.12	13.49	27.2	54.37	1.78	6.38	13.7	26.15	52.62
Foreman	12.87	28.92	58.2	5.7	13.58	27.1	54.35	1.76	6.31	13.14	26.16	52.64
Soccer	12.74	29.1	58.24	5.4	13.52	27.4	54.4	1.71	6.43	13.1	26.22	52.64
Harbour	12.96	28.89	58.16	5.14	13.5	27	54.36	1.78	6.36	13.7	26.14	52.64
Mobile	12.88	28.91	58.21	5.9	13.53	27.2	54.37	1.75	6.39	13.9	26.16	52.62
Average	12.89	28.92	58.19	5.1	13.52	27.2	54.37	1.76	6.37	13.8	26.16	52.64

Table 2 Encoding time (%) for each temporal layer (TL) with different GOP sizes using CIF Baseline Profile

Encoding time (%) of every temporal layer CIF (30 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	TL0	TL1	TL2	TL0	TL1	TL2	TL3	TL0	TL1	TL2	TL3	TL4
Hall	12.71	29.04	58.25	5.09	13.46	27.09	54.36	2.45	6.46	12.91	26	52.18
City	12.68	29.04	58.28	5.05	13.52	27.15	54.29	2.45	6.47	12.93	26.05	52.13
Foreman	12.55	29.09	58.36	5.09	13.5	27.09	54.32	2.44	6.49	12.95	25.98	52.13
Soccer	12.72	29.03	58.25	5.01	13.5	27.14	54.35	2.38	6.45	12.98	26.04	52.15
Harbour	12.55	29.09	58.36	5.12	13.45	27.09	54.34	1.6	6.51	13.03	26.23	52.62
Mobile	12.67	29.05	58.28	5.08	13.47	27.1	54.36	2.42	6.48	12.93	26.01	52.17
Average	12.65	29.06	58.3	5.07	13.48	27.11	54.34	2.29	6.48	12.96	26.05	52.23

Table 3 Encoding time (%) for each temporal layer (TL) with different GOP sizes using QCIF Main Profile

Encoding time (%) of every temporal layer QCIF (15 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	TL0	TL1	TL2	TL0	TL1	TL2	TL3	TL0	TL1	TL2	TL3	TL4
Hall	12.86	28.92	58.23	5.08	13.52	27.1	54.3	1.77	6.36	13.07	26.14	52.65
City	12.87	28.94	58.19	5.1	13.51	27.02	54.37	1.78	6.38	13.07	26.15	52.65
Foreman	12.56	29.64	57.8	4.71	13.73	27.32	54.24	1.76	6.31	13.14	26.16	52.64
Soccer	12.7	29.02	58.28	4.99	13.44	27.13	54.44	1.71	6.43	13.01	26.22	52.64
Harbour	12.91	28.92	58.17	5.13	13.54	27	54.33	1.78	6.36	13.07	26.14	52.64
Mobile	12.82	28.94	58.24	4.72	13.65	27.37	54.26	1.75	6.39	13.09	26.14	52.62
Average	12.79	29.06	58.15	4.96	13.57	27.16	54.32	1.76	6.37	13.08	26.16	52.64

doubles when QP increases by 6, the thresholds are adjusted by 12.5 % for a change in QP of 1.

4.2 Reducing motion estimation complexity

As said previously, the idea of the motion estimation task consists of eliminating the temporal redundancy in a way that corresponds to the movement of the scene. For this purpose, in H.264/AVC MVs between every MB or sub-MB partition and the block which generates the lowest residual inside the search area of the reference frame are calculated. These MVs represent, approximately, the amount of movement of the MB. Since the MVs, generated by H.264/AVC and transmitted into the encoded bitstream, represent, approximately, the amount of movement of the frame, they can be reused to accelerate the SVC motion estimation process by reducing the search area dynamically and efficiently. The new reduced search area which is proposed uses the incoming MVs from H.264/AVC to determine a small area to find the real MVs calculated in SVC, which is depicted in Fig. 6.

Table 4 Encoding time (%) for each temporal layer (TL) with different GOP sizes using CIF Main Profile

Encoding time (%) of every temporal layer CIF (30 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	TL0	TL1	TL2	TL0	TL1	TL2	TL3	TL0	TL1	TL2	TL3	TL4
Hall	13.53	28.78	57.69	5.93	13.34	26.85	53.88	1.57	6.56	13.15	26.39	52.33
City	13.51	28.78	57.71	5.93	13.34	26.86	53.87	2.43	6.46	12.92	26.01	52.18
Foreman	13.46	28.81	57.73	5.91	13.34	26.86	53.88	1.52	6.63	13.11	26.34	52.4
Soccer	13.35	28.84	57.81	5.84	13.34	26.89	53.93	1.54	6.55	13.11	26.35	52.45
Harbour	13.51	28.77	57.72	5.94	13.54	26.85	53.86	2.44	6.46	12.92	26.01	52.17
Mobile	13.52	28.83	57.64	5.91	13.34	26.86	53.9	1.49	6.61	13.09	26.32	52.5
Average	13.48	28.8	57.72	5.91	13.34	53.89	53.89	1.83	6.55	13.05	26.24	52.34

Table 5 RD performance and time savings of the approach for GOP = 2, different resolutions and Baseline Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 2 − Baseline Profile								
QCIF (15 Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.044	−0.11	66.94	99.12	0.055	−0.11	66.1	98.85
City	0.026	0.91	66.88	99.04	0.056	0.21	66	98.96
Foreman	0.078	1.17	65.54	97.19	−0.058	1.45	65.21	97.67
Soccer	0.37	1.45	63.83	94.42	0.022	1.22	63.85	95.58
Harbour	0.027	−0.34	66.1	98.36	0.053	−0.57	64.15	96.95
Mobile	0.041	−0.4	66.97	98.26	0.092	−1.55	65.81	98.15
Average	0.042	0.45	66.04	97.73	0.037	0.11	65.19	97.69

Table 6 RD performance and time savings of the approach for GOP = 4, different resolutions and Baseline Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 4 − Baseline Profile								
QCIF (15 Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.222	−0.02	85.91	99.14	0.331	−0.53	86.64	99.08
City	0.066	1.87	86.13	99.11	0.204	0.59	87.23	99.24
Foreman	0.259	2.16	83.25	97.01	−0.108	2.92	84.64	97.7
Soccer	0.037	2.51	81.77	94.52	0.022	2.3	82.58	95.6
Harbour	0.112	−0.82	85.38	98.44	0.181	−1.43	87.3	98.87
Mobile	0.151	−0.17	84.33	98.19	0.246	−2.3	85.24	98.24
Average	0.141	0.92	84.46	97.74	0.146	0.26	85.61	98.13

This smaller search area 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 length of the average of the incoming vector for a specific MB and the temporal layer which the frame is in. Both 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 (bidirectional) for prediction. For P pictures only list0 is used.

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 the reference from the list1 or vice versa or even a bidirectional prediction is done requiring MVs of both lists.

Table 7 RD performance and time savings of the approach for GOP = 8, different resolutions and Baseline Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 8 − Baseline Profile								
QCIF (15Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.159	0.35	80	98.74	0.026	0.45	79.84	98.87
City	−0.003	2.61	80.02	99.12	0.178	1.25	79.83	99.04
Foreman	0.219	3.03	77.29	96.89	0.0005	3.48	78.78	97.73
Soccer	0.066	2.96	75.45	94.49	0	2.55	76.96	95.67
Harbour	0.052	0.02	78.63	98.4	0.077	0.38	79.46	98.37
Mobile	0.038	0.57	79.24	98.36	0.248	−1.37	79.31	98.34
Average	0.089	1.59	78.44	97.67	0.089	1	79.03	98

Table 8 RD performance and time savings of the approach for GOP = 16, different resolutions and Baseline Profile

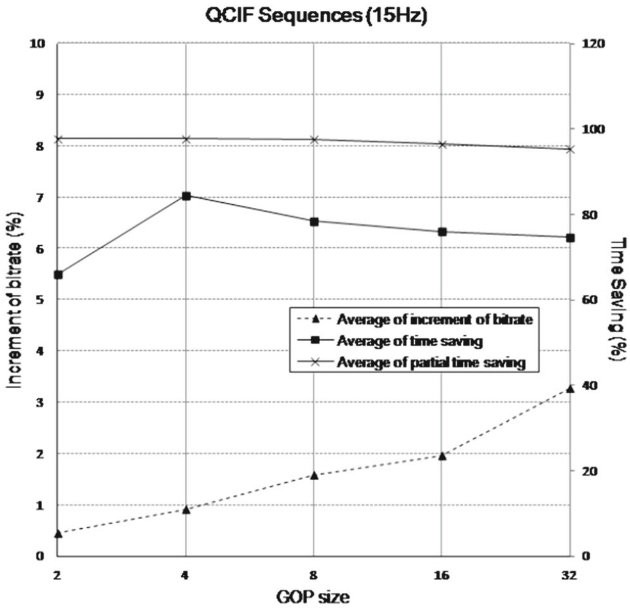
RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 16 − Baseline Profile								
QCIF (15Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.237	0.52	77.63	97.86	−0.671	1.65	76.17	98.99
City	−0.035	3.06	77.54	97.87	−0.138	1.9	76.22	99.1
Foreman	0.088	3.12	73.98	95.49	−0.097	4.78	75.06	97.65
Soccer	0.063	3.32	73.81	93.55	0.032	3.66	73.43	95.71
Harbour	0.204	0.86	77.34	97.27	0.285	−2.6	75.68	98.46
Mobile	0.03	0.91	76.41	97.07	0.232	−0.39	75.93	98.45
Average	0.113	1.97	76.12	95.62	−0.06	1.5	75.42	98.06

Table 9 RD performance and time savings of the approach for GOP = 32, different resolutions and Baseline Profile

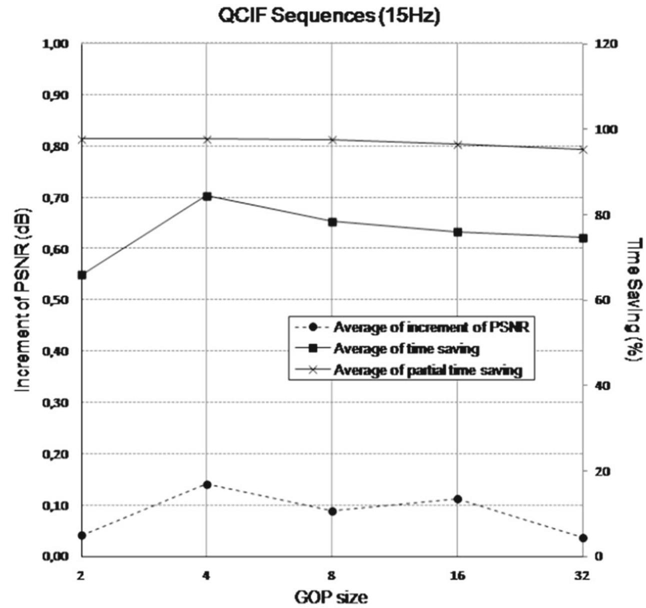
RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 32 − Baseline Profile								
QCIF (15 Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.293	1.16	75.6	92.09	0.757	1.11	74.94	98.22
City	−0.19	3.53	76	91.98	−0.101	2.68	75.01	98.44
Foreman	−0.11	5.35	73.77	96.53	−0.26	5.17	73.96	97.07
Soccer	0.057	5.35	72.34	94.82	0.017	4.23	72.38	95.05
Harbour	0.126	2.29	75.3	98.52	−0.005	2.31	74.76	97.88
Mobile	0.045	2	75.01	98.48	0.171	1.13	74.69	97.84
Average	0.037	3.28	74.67	95.4	0.097	2.77	74.29	97.42

In these cases, the supposition is made that the length of the MV of both lists for a MB is the same.

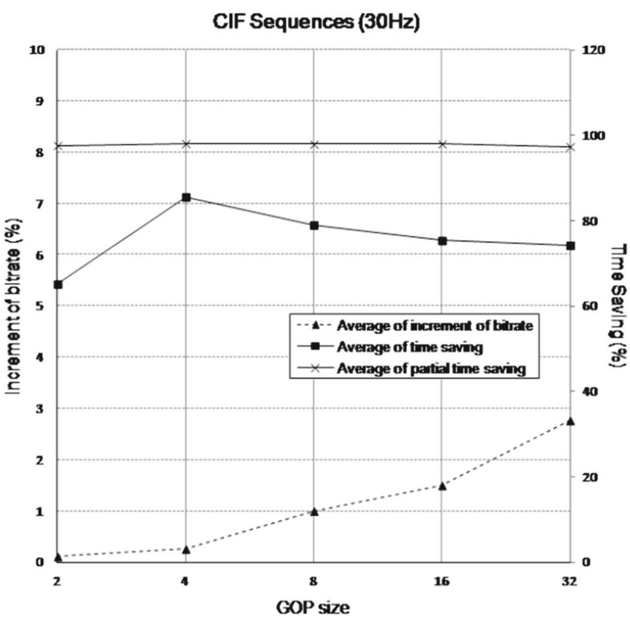
The average of the incoming MVs of a determined MB is used to overcome the mismatching between GOP patterns and prediction structures. While the starting encoded bitstream in H.264/AVC is formed by IPPP/IBBP GOP patterns without temporal scalability, the final SVC bitstream needs temporally scalable hierarchical structures (see Fig. 2). This fact leads to different MVs in both H.264/AVC and SVC. Furthermore, MB partitions developed by H.264/AVC can be different from those in SVC (see Fig. 7), so the number of MVs associated with an H.264/AVC MB can be different



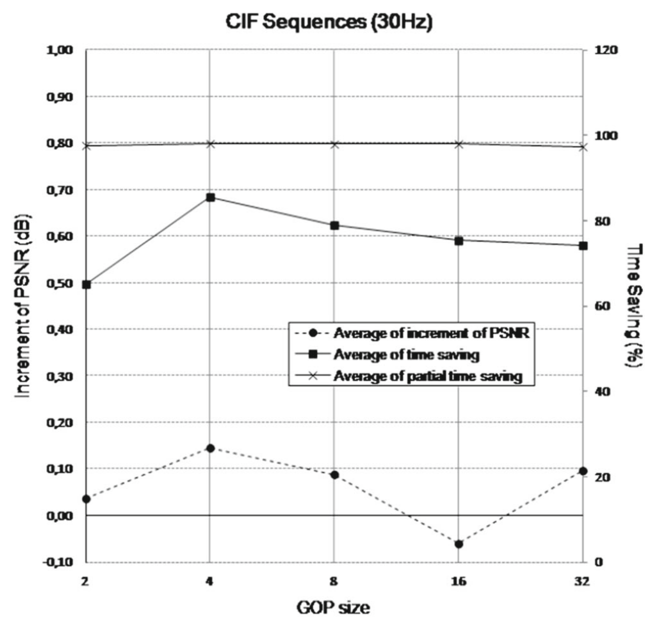
(a) $\Delta Bitrate$ vs. GOP size vs. Time Saving QCIF resolution



(b) Loss of PSNR vs.GOP size vs. Time Saving QCIF resolution



(c) $\Delta Bitrate$ vs. GOP size vs. Time Saving CIF resolution



(d) Loss of PSNR vs. GOP size vs. Time Saving CIF resolution

Fig. 8 Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions Baseline Profile

from the number of MVs associated with the corresponding SVC. The present approach tries to tackle this problem.

Another thing to keep in mind is that these MVs for each MB have been calculated in H.264/AVC using a reference frame that could have a different distance from the current frame than for the SVC case. In general, GOP structures in SVC with temporal scalability lead to longer distances between a frame and its reference frame than in H.264/AVC. As can be seen in Fig. 2, with hierarchical picture structures, the distance between both frames is longer when the temporal layer decreases. To deal with this different prediction distance, a correction factor is introduced so the circumference generated previously is multiplied by a factor that depends on which temporal layer the current frame is in.

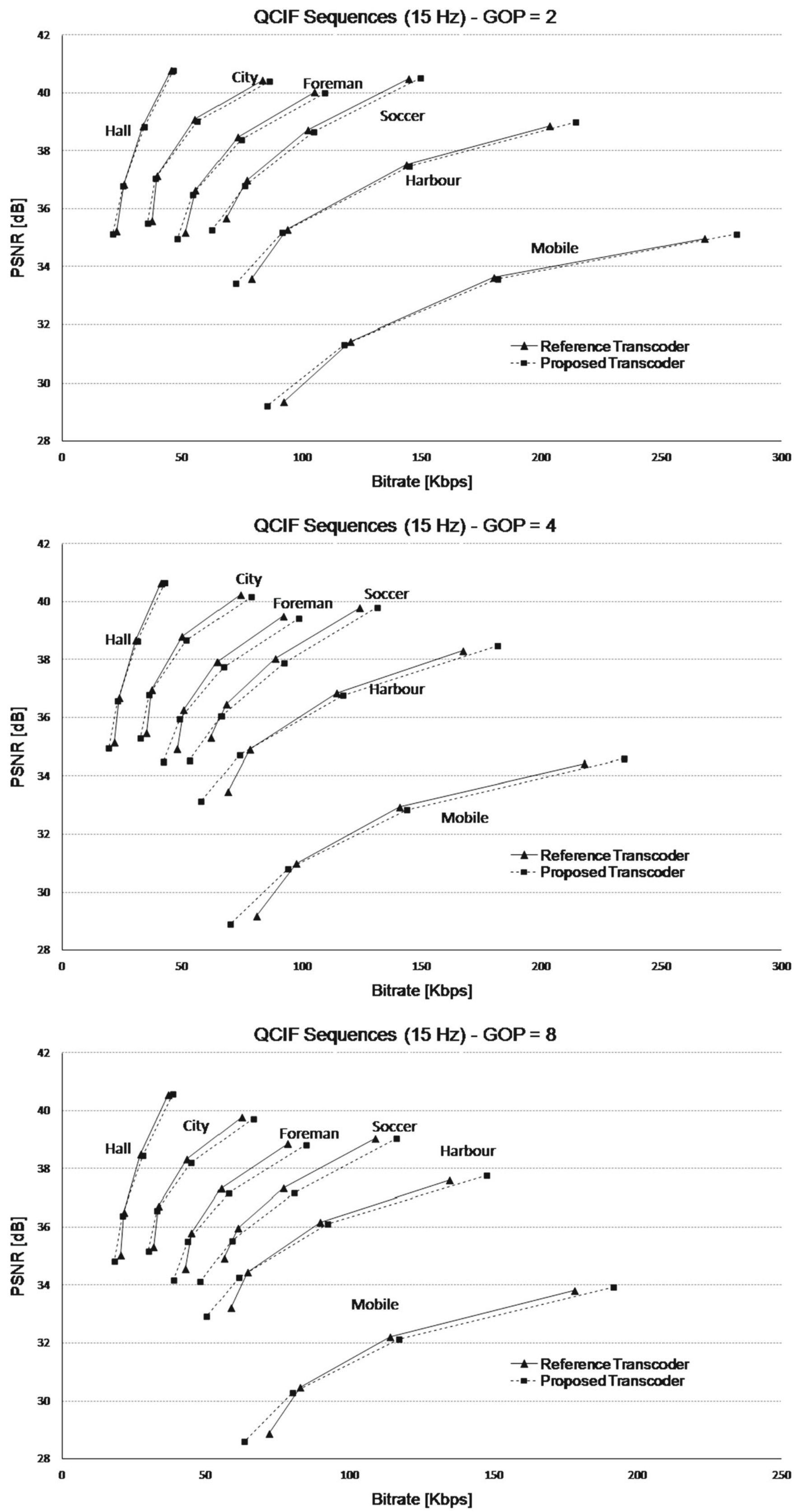


Fig. 9 RD performance in QCIF resolution with different GOP sizes Baseline Profile

Fig. 10 RD performance in CIF resolution with different GOP sizes Baseline Profile

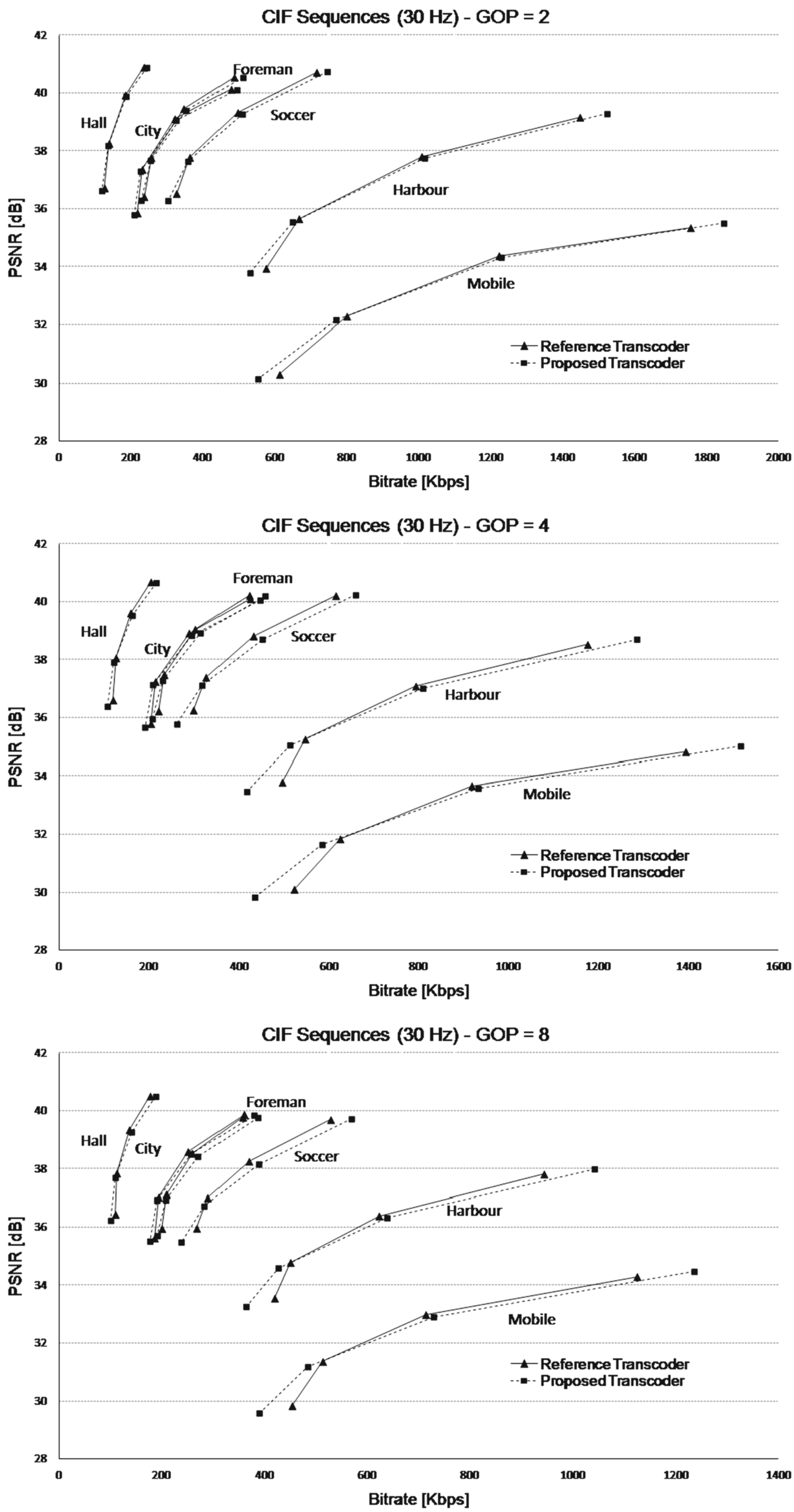


Table 10 RD performance and time savings of the approach for GOP = 2, different resolutions and Main Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 2 − MainProfile								
QCIF (15 Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.065	−0.43	67.93	99.2	0.134	−1.18	65.63	99.13
City	0.06	0.34	68.8	99.12	0.049	−0.44	65.61	99.18
Foreman	−0.005	1.57	65.63	96.99	−0.041	1.52	64.84	97.62
Soccer	−0.023	3.75	65.05	99.45	−0.01	1.94	63.48	95.42
Harbour	0.07	−0.6	68.74	98.86	0.071	−0.55	65.51	98.78
Mobile	0.018	−0.2	68.8	99.01	0.06	−0.97	65.39	98.83
Average	0.031	0.74	67.49	98.77	0.044	0.05	65.08	98.16

Table 11 RD performance and time savings of the approach for GOP = 4, different resolutions and Main Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 4 − MainProfile								
QCIF (15Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.168	0.12	87.02	99.27	0.231	−1.33	85.27	99.15
City	0.143	2.03	86.47	99.12	0.019	0.26	85.21	99.19
Foreman	0.046	3.57	84.67	99.71	−0.02	3.33	83.96	97.65
Soccer	−0.064	4.93	81.8	99.45	0.067	3.62	82.07	95.52
Harbour	0.175	−0.56	86.24	98.9	0.141	−1.01	85	98.83
Mobile	0.047	0.57	86	99.01	0.106	−0.99	85.95	99.14
Average	0.086	1.78	85.37	99.24	0.091	0.65	84.58	98.25

5 Performance evaluation

In this section, results from the implementation of the proposal described in the previous section are shown. Experiments were conducted to evaluate the performance of the proposed H.264/AVC-to-SVC transcoder when transcoding videos using test sequ-ences of varying characteristics, namely Hall, City, Foreman, Soccer, Harbour, and Mobile in CIF (30 Hz) and QCIF resolutions (15 Hz).

These sequences were encoded using the H.264/AVC Joint Model (JM) reference software [17], version 16.2, with an IPPP/IBBP patterns with a fixed QP = 28 in a trade-off between quality and bitrate. Then, for the reference results, the encoded bit-streams are decoded and re-encoded using the JSVM software, version 9.19.3 [18] with temporal scalability, Baseline and Main Profiles and different values of QP (28, 32, 36, 40) and different GOP sizes.

For the results of our proposal, encoded bitstreams in H.264/AVC are transcoded using the technique described in Sect. 4. This technique was applied to the two enhancement temporal layers with the highest identifiers because, as it was shown in Tables 1, 2, 3, and 4, these temporal layers were where most encoding time is spent (approximately 80 % of the time spent on encoding the full sequence is used to encode these temporal layers). If there is only one temporal enhancement layer, it will only be applied to this one to avoid changes in the base temporal layer.

The metrics used to evaluate the proposed video transcoder are the RD function (Bitrate vs. PSNR), $\Delta Bitrate$ (%), $\Delta PSNR$ (dB) and Time Saving (%). These metrics are defined below:

– *RD function* Rate distortion gives theoretical bounds to the compression rates that can be achieved using different methods. In rate distortion theory, the rate is usually understood as the number of bits per data sample to be stored or transmitted. The notion of distortion is a subject of ongoing discussion. In the simplest case (which is actually used in most cases), the distortion is defined as the variance of the difference between the input and the output signals (i.e., the mean squared error of the difference). In the definition of the RD function used

Table 12 RD performance and time savings of the approach for GOP = 8, different resolutions and Main Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 8 − MainProfile								
QCIF (15Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.666	−0.03	81.29	99.29	0.444	−1.12	79.34	99.16
City	0.065	1.75	79.89	99.11	0.02	0.23	79.31	99.2
Foreman	0.04	3.39	79.4	99.71	−0.175	3.15	78.14	97.68
Soccer	−0.016	4.97	75.72	99.43	0.094	4.1	76.38	95.55
Harbour	0.366	−0.66	80.29	98.9	0.248	−0.89	79.16	98.88
Mobile	0.026	0.52	80.73	99.09	0.219	−0.53	81.15	99.22
Average	0.191	1.66	79.55	99.26	0.142	0.82	78.91	98.28

Table 13 RD performance and time savings of the approach for GOP = 16, different resolutions and Main Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 16 − MainProfile								
QCIF (15 Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.632	−0.06	76.12	97.8	0.337	−0.9	76.38	99.27
City	−0.007	1.87	75.42	97.62	−0.006	0.29	76.32	99.22
Foreman	0.096	2.85	73.21	95.44	−0.106	3.23	75.14	97.62
Soccer	−0.059	4.41	71.47	92.83	0.105	4.33	73.54	95.6
Harbour	0.175	−0.52	75.18	97.41	0.171	−0.67	76.26	98.92
Mobile	0.038	0.33	75.33	97.58	0.192	−0.18	77.75	99.25
Average	0.146	1.48	74.46	96.45	0.116	1.02	75.9	98.31

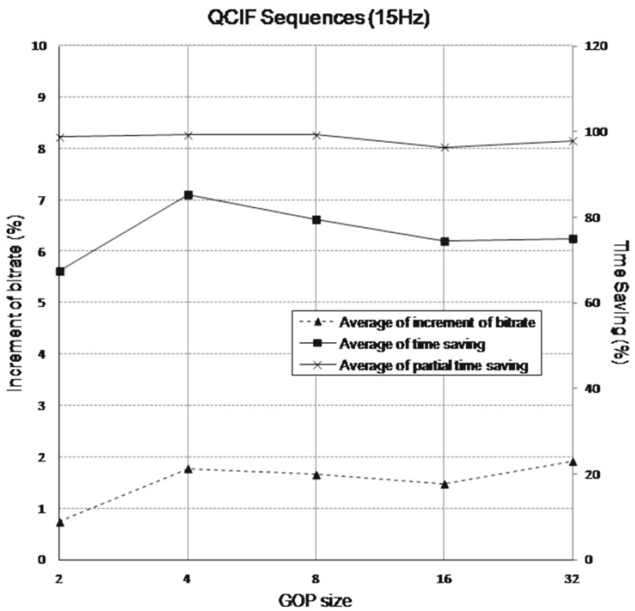
Table 14 RD performance and time savings of the approach for GOP = 32, different resolutions and Main Profile

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 32 − MainProfile								
QCIF (15 Hz)					CIF (30 Hz)			
Sequence	ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)		ΔPSNR (dB)	ΔBitrate (%)	Time saving (%)	
			Full seq.	Partial			Full seq.	Partial
Hall	0.627	0.29	76.73	99.26	0.167	−0.34	74.88	98.57
City	−0.019	1.63	76.07	99.17	0.1	0.4	75.09	98.54
Foreman	−0.024	3.5	74.42	96.7	−0.002	3.4	74.02	97.03
Soccer	−0.013	5.23	71.4	96.44	0.076	4.38	72.37	95
Harbour	0.374	−0.02	76.3	98.96	0.231	−0.26	75.06	98.28
Mobile	0.107	0.86	75.71	99.16	0.15	0.19	75.75	98.58
Average	0.175	1.92	75.11	97.95	0.12	1.3	74.53	97.67

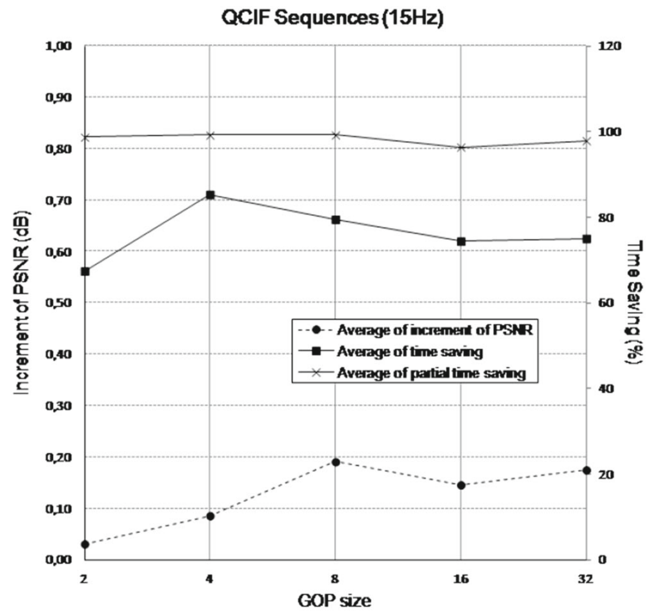
to show the performance results, PSNR is the distortion for a given bitrate. The averaged PSNR values of luminance (Y) and chrominance (U, V) are used in the RD function graphs. The averaged-global PSNR is based on Eq. 1.

$$\overline{PSNR} = \frac{4PNSR_I + PSNR_U + PSNR_V}{6} \quad (1)$$

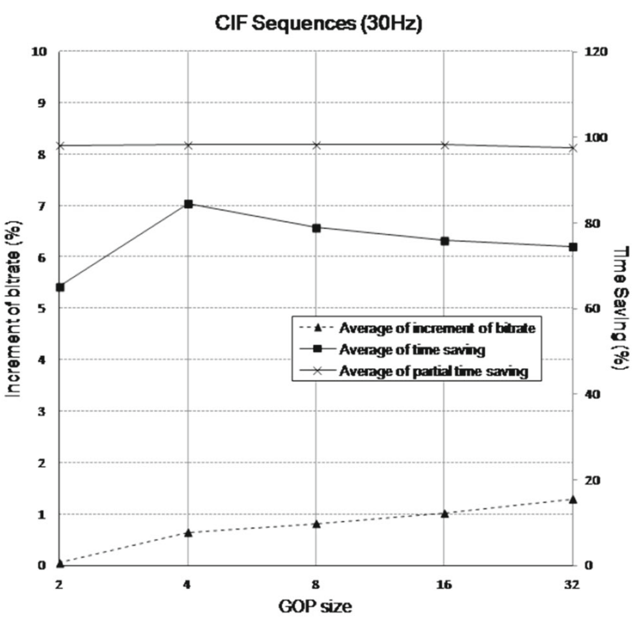
– $\Delta PSNR$ **(dB) and** $\Delta Bitrate$ (%): The detailed procedures for calculating these differences can be found in a



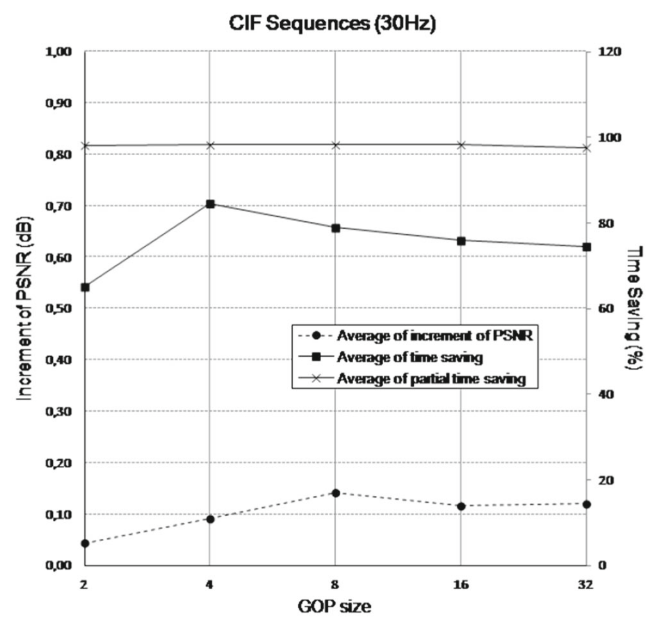
(a) $\Delta Bitrate$ vs. GOP size vs. Time Saving QCIF resolution



(b) Loss of PSNR vs.GOP size vs. Time Saving QCIF resolution



(c) $\Delta Bitrate$ vs. GOP size vs. Time Saving CIF resolution



(d) Loss of PSNR vs. GOP size vs. Time Saving CIF resolution

Fig. 11 Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions Main Profile

JVT document authored by Bjntegaard [24]. This mechanism is proposed for finding numerical averages between RD-curves as part of the presentation of the results. $\Delta PSNR$ represents the difference in quality (negative means quality loss) and $\Delta Bitrate$ repre-sents the bitrate increment (positive means that the bitrate increases).

– *Time Saving* (%) In order to evaluate the complexity reduction achieved by the proposal compared to the reference transcoder, the following calculation is defined to find the time differences. Let Tref denote the coding time used by the H.264/AVC reference software and Tprop be the time taken by the algorithm proposed or the mecha-

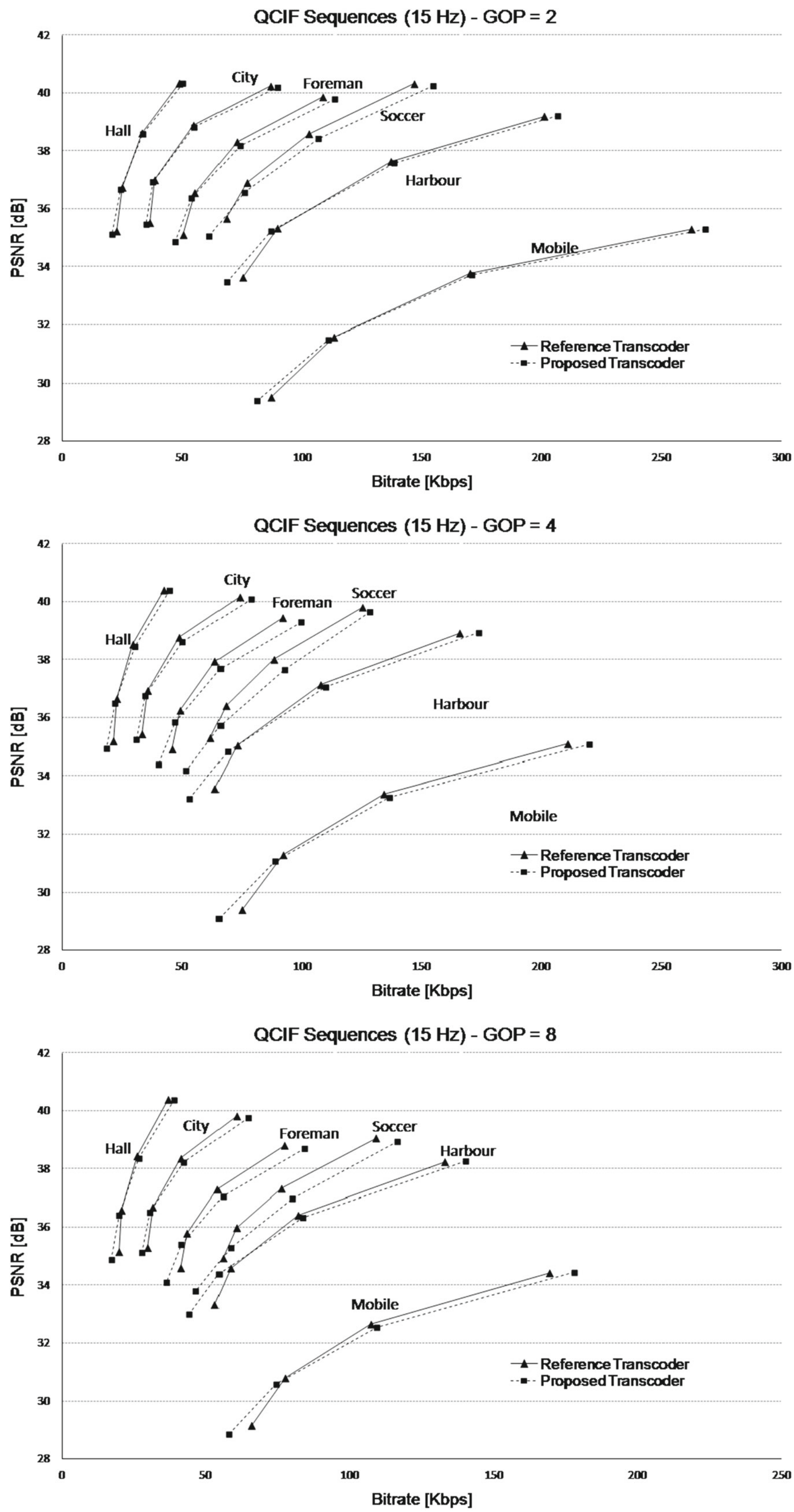


Fig. 12 RD performance in QCIF resolution with different GOP sizes Main Profile

nism that has been evaluated; Time Saving is defined in Eq. 2. In Tprop the full computational cost for the operations needed to prepare the information for the approach is also included. In the proposal presented in this paper, there are two different Time Savings calculated:

- *Full seq.* This is the time reduction for the whole sequence when our proposal is applied.
- *Partial* This is the time reduction for the temporal layers which the proposal is applied to.

$$TimeSaving(\%) = \frac{T_{ref} - T_{prop}}{T_{ref}} * 100 \qquad (2)$$

5.1 Baseline profile evaluation

This section discusses the performance evaluation of the proposed H.264/AVC-to-SVC transcoder for Baseline Profile. In Table 5, 6, 7, 8, and 9 the results for $\Delta PSNR$, $\Delta Bitrate$ and Time Saving are shown when our technique is applied compared to the reference transcoder. Moreover, these results are collected graphically in Fig. 8.

The values of PSNR and bitrate obtained with the proposed transcoder are very close to the results obtained when applying the reference transcoder (re-encoder) while around 80–85 % of reduction of computational complexity in the full sequence and around 98 % in the specific layers is achieved for Baseline Profile. Some resulting rate–distortion (RD) curves for the SVC bitstreams with several GOP sizes are shown in Figs. 9 and 10 where it can be seen that our proposal for transcoding is able to approach the RD-optimal transcoded (re-encoded) reference without any significant loss. The values of PSNR and bitrate obtained with the proposed transcoder are ve-ry close to the results obtained when applying the reference transcoder (re-encoder) while a significant reduction of computational complexity is achieved (around a 98 % where the proposal is applied).

5.2 Main Profile evaluation

This section discusses the performance evaluation of the proposed H.264/AVC-to-SVC transcoder for Main Profile. For measuring the results of the proposal, the same test sequences used previously in QCIF and CIF resolution were used. The results for running the technique for different GOP sizes are shown in Tables 10, 11, 12, 13, and 14. Moreover, these results are collected graphically in Fig. 11. Some resulting RD curves for the SVC bitstreams with several GOP sizes are shown in Figs. 12 and 13 where it can be seen that our proposal for transcoding is able to approach the RD-optimal transcoded (re-encoded) reference without any significant loss.

5.3 Analysis of results

In this section, an analysis of the results obtained is done. Both in Baseline and Main Profile time reduction has been increased when the proposals presented in Sect. 4 have been adjusted for working together. In Baseline Profile around a 97 % of time saving in the temporal layers where the proposal is applied is achieved, while around a 75 % of time saving has been achieved in the whole sequence. In Main Profile the time saving achieved is slightly higher. These reductions were obtained with an increment of bitrate of 3.28 % in the worst case. Regarding the loss of quality, the proposal is able to improve the quality of the reference. This is possible because the reference was encoded with RDO option disable. As in the previous proposals, the results show that the technique can be applied to different GOP sizes obtaining results very similar.

The performance results also show that the proposal works properly with different sequences with varying characteristics and resolutions, although there are some differences between sequences with regard to the increment in bitrate. For example, the increment in bitrate is smaller in Hall or Harbour than in Soccer. This is due to the high level of movement in the Soccer sequence. Since the prediction structure in H.264/AVC without scalability and SVC is different, the reference frames from the same frame number are usually different. As the information collected from the decoding stage for each frame (residual, MVs, mode decision) is used for the decision tree for deciding the MB type, if the scene has little movement, the different prediction structure has less impact than if the sequence has a high level of movement.

Another thing that can be observed is that the proposal can be applied to different GOP sizes and the results are very similar in all cases. Only in one case, when the transcoding techniques are applied to sequences encoded with a GOP size of 4, does the time reduction achieve its maximum value. This is due to the fact that in this case the techniques are applied only to two out of three temporal layers and only the temporal base layer is encoded completely. In conclusion, the proposal can be applied to different GOP sizes and works properly in all of them.

6 Conclusion

In this paper, a proposal for adapting H.264/AVC bitstreams to SVC streams with temporal scalability in Baseline and Main Profile has been presented. This scalability makes it possible to adapt the video contents to different mobile devices regarding frame rate. Moreover, by applying our proposal, the complexity of the interprediction process is reduced, and therefore, the complexity of the adaptation. The experimental results show that it is capable to reduce

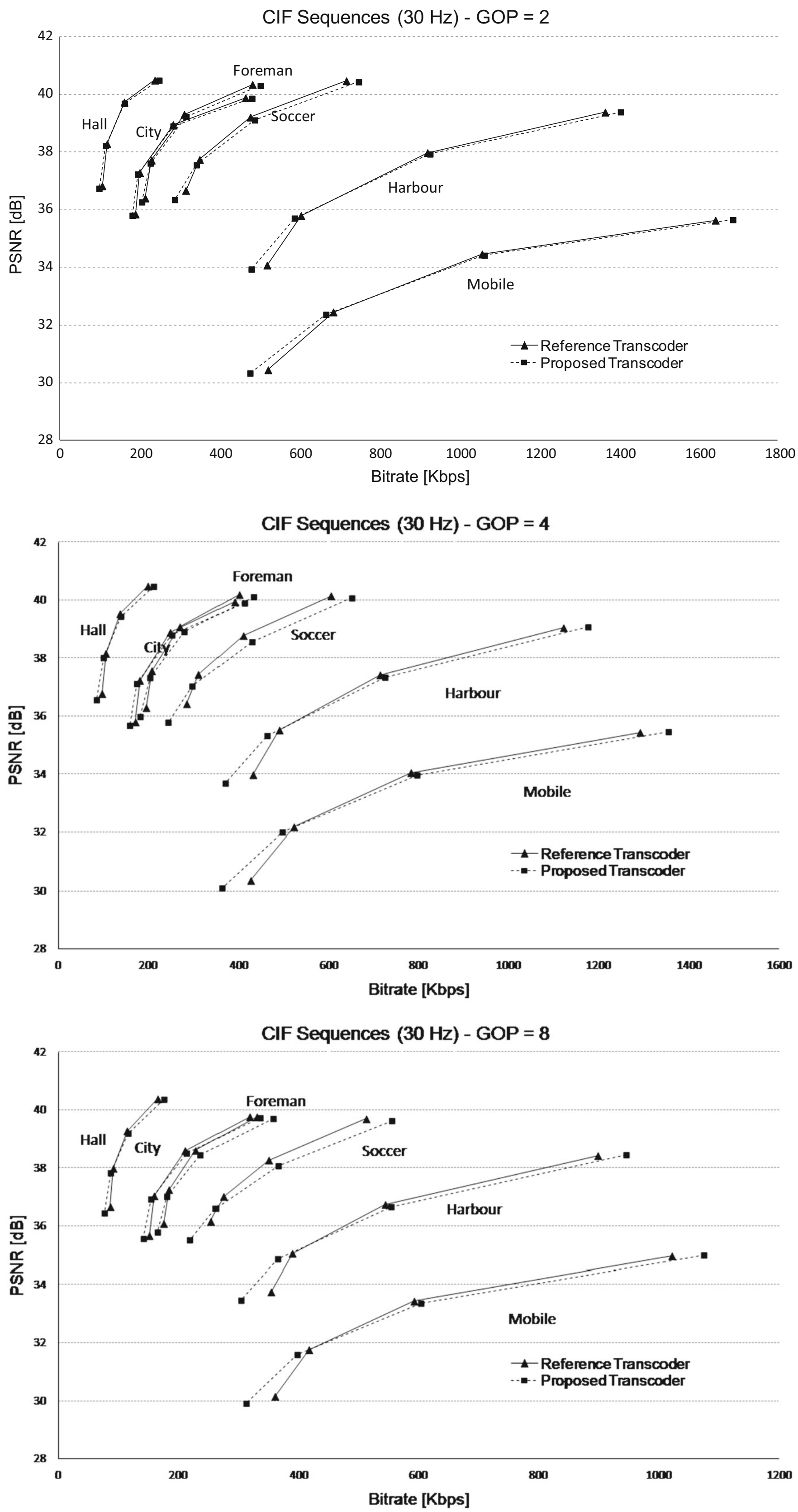


Fig. 13 RD performance in CIF resolution with different GOP sizes Main Profile

the coding complexity by around 98 % where it is applied while maintaining the coding efficiency.

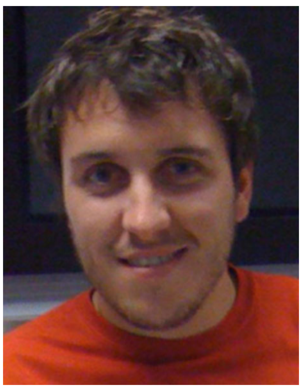
Acknowledgments This work was supported by the Spanish MEC and MICINN funds, under the Grant TIN2012-38341-C04-04. The first author would also like to thank Spanish Public Employment Service for its funding support.

References

1. Advanced Television System Committee: ATSC-Mobile DTV Standard, A/153 ATSC Mobile Digital Television System. October 2009.
2. Al-Muscati, H., & Labeau, F. (2010). Temporal Transcoding of H.264/AVC Video to the Scalable Format. *2nd international conference on image processing theory tools and applications*, Paris.
3. Chia-Hung, Y., Wen-Yu, T., & Shih-Tse, W. Mode decision acceleration for H.264/AVC to SVC temporal video transcoding, *EURASIP Journal on Advances in Signal Processing*. doi:10.1186/1687-6180-2012-204.
4. Cohen, W. (1995). Fast Effective Rule Induction. *20th international conference on machine learning*, pp. 115–123.
5. De Cock, J., Notebaert, S., Lambert, P., & Van de Walle, R. (2009). Architectures of fast transcoding of H.264/AVC to quality-scalable SVC streams. *IEEE Transaction on Multimedia*, *11*(7), 1209–1224.
6. Develder, C., Lambert, P., Van Lancker, W., et al. (2012). Delivering scalable video with QoS to the home. *Telecommunication Systems*, *49*(1), 129–148.
7. Dziri, A., Diallo, A., Kieffer, M., & Duhamel, P. (2008). P-Picture based H.264 AVC to H.264 SVC temporal transcoding, international wireless communications and mobile computing conference.
8. European Broadcasting Union: ETSI TR 102 377 V1.4.1: Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines. June 2009.
9. Garrido-Cantos, R., De Cock, J., Martínez, J. L., Van Leuven, S., & Garrido, A. (2013). Video transcoding for mobile digital television. *Telecommunication Systems*, 52(4), 2655–2666.
10. Garrido-Cantos, R., De Cock, J., Martnez, J. L., Van Leuven, S., & Cuenca, P. (2011). Motion-based temporal transcoding from H.264/AVC-to-SVC in Baseline profile. *IEEE Transactions on Consumer Electronics*, *57*(1), 239.
11. Garrido-Cantos, R., De Cock, J., Martnez, J. L., Van Leuven, S., Cuenca, P., Garrido, A., & Van de Walle, R. (2011). Low complexity adaptation for mobile video environments using data mining, *4th IFIP wireless and mobile networking conference* (WMNC 2011).
12. Garrido-Cantos, R., De Cock, J., Martnez, J. L., Van Leuven, S., Cuenca, P., & Garrido, A. (2014). Scalable video transcoding for mobile communications. *Telecommunication Systems*, 55(2), 173–184.
13. Garrido-Cantos, R., De Cock, J., Martnez, J. L., Van Leuven, S., Cuenca, P., Garrido, A., & Van de Walle, R. (2011). Temporal video transcoding for digital TV broadcasting, *5th IFIP wireless and mobile networking conference* (WMNC 2012), Bratislava.
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. The WEKA data mining software: An update. *SIGKDD Explorations*, *11*(1), 21–35.
15. 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.
16. Lian, S. Secure service convergence based on scalable media coding. *Telecommunication Systems*, *45*, 1.
17. Joint Model JM Reference Software. http://iphome.hhi.de/suehring/tml/download/.
18. Joint Scalable Video Model (JSVM) Reference Software.http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm.
19. Monteiro, J. M., Calafate, C., & Nunes, M. (2012). Robust multipoint and multi-layered transmission of H.264/SVC with Raptor codes. *Telecommunication Systems*, *49*(1), 113–128.
20. Sachdeva, R., Johar, S., & Piccinelli, E. (2009). Adding SVC spatial scalability to existing H.264/AVC video, *8th IEEE/ACIS international conference on computer and information science*, Shangai.
21. Schwarz, H., Marpe, D., & Wiegand, T. (2006). Analysis of hierarchical B pictures and MCTF. *IEEE international conferene on ICME and expo*, Toronto.
22. Schwarz, H., Marpe, D., & Wiegand, T. (2007). Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *17*(9), 1103–1120.
23. Soohong, P., & Seong-Ho, J. (2009). Mobile IPTV: Approaches, challenges, standards and QoS support. *IEEE Internet Computing*, *13*(3), 23–31.
24. Sullivan, G., & Bjntegaard, G. (2001). Recommended S-mulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material. ITU-T VCEG, Doc. VCEG-N81. September.
25. Van Leuven, S., De Cock, J., Van Wallendael, G., Van de Walle, R., Garrido-Cantos, R., Martinez J. L., & Cuenca, P. A Low-complexity closed-loop H.264/AVC to quality-scalable SVC transcoder, 7th international conference on digital signal processing.
26. Van Leuven, S., De Cock, J., Van Wallendael, G., Van de Walle, R., Garrido-Cantos, R., Martinez, J.L., & Cuenca, P. (2011). Combining open- and closed-loop architectures for H.264/AVC-to-SVCtranscoding, *Proceedings of 18th IEEE international conference on image processing*, pp. 1661–1664, Brussels, Belgium.
27. Van Wallendael, G., Van Leuven, S., Garrido-Cantos, R., De Cock, J., Martinez, J.L., Lambert, P., Cuenca, P., & Van de Walle, R. (2010). Fast H.264/AVC-to-SVC transcoding in a mobile television environ-ment, *Mobile multimedia communications conference, 6th international ICST, Proceedings*, Lisbon.
28. Vetro, A., Christopoulos, C., & Sun, H. (2003). Video transcoding architectures and techniques: An overview. *IEEE Signal Processing Magazine*, *20*(2), 18–29.
29. Wenger, S. (1998). Temporal scalability using P-pictures for low-latency applications. *IEEE Second Workshop on Multimedia Signal Processing* (p. 559564). Redondo Beach: CA, USA.

R. Garrido-Cantos received the M.Sc. degree in Telecommunication Engineering from European University of Madrid, Spain, in 2007, and the M.Sc. and Ph.D. degrees in Computer Science from University of Castilla-La Mancha, Spain, in 2010 and 2012, respectively. From 2004 to 2008 she worked in several companies such as Telefónica I + D, Nokia Siemens Networks, and Accenture. In 2009 she joined the Albacete Research Institute of Informatics, University of Castilla- La Mancha, where she has been working towards a Ph.D. Her research interests include scalable video coding, and video transcoding. She has also been an exchange student at Cork Institute of Technology, Ireland and a visiting researcher at Ghent University, Belgium.

J. De Cock obtained the M.S. and Ph.D. degrees in Engineering from Ghent University, Belgium, in 2004 and 2009, respectively. Since 2004 he has been working at Multimedia Lab, Ghent University, and the Interdisciplinary Institute for Broadband Technology (IBBT). His research interests include video compression and transcoding, scalable video coding, and multimedia applications.



J. L. Martinez received his M.S and Ph.D. degrees in Computer Science and Engineering from the University of Castilla-La Mancha, Spain in 2007 and 2009 respectively. In 2005, he joined the Department of Computer Engineering at the University of Castilla-La Mancha, where he was a researcher. In 2010, he joined the department of Computer Architecture of Complutense University in Madrid where he was assistant professor. Currently he is assistant professor in the University of Castilla-LaMancha. His research interests include Distributed Video Coding (DVC), multimedia standards, video transcoding, parallel video processing. He has also been a visiting researcher at the Florida Atlantic University (USA) and Centre for Communication System Research at the University of Surrey (UK). He has over 40 publications in these areas in international refereed journals and conference proceedings.

S. Van Leuven received the M.Sc. degree in Applied Engineering from the University College of Antwerp, Antwerp, Belgium, in 2006 and the M.Sc. degree in Computer Science Engineering from Ghent University, Ghent, Belgium in 2008. Currently, he is with Multimedia Lab, Ghent University, where he is working towards a Ph.D., with the financial support of the Agency for Innovation by Science and Technology (IWT). His main research topic is video coding, including scalable video coding, adaptation of video streams, transcoding and next generation video compression.

P. Cuenca received his M.Sc. degree in Physics (award extraordinary) from the University of Valencia in 1994. He got his Ph.D. degree in Computer Engineering in 1999 from the Polytechnic University of Valencia. In 1995 he joined the Department de Computer Engineering at the University of Castilla-La Mancha. He is currently a Full Professor of Communications and Computer Networks and Dean of the Escuela Superior de Ingeniería Informática in Albacete. His research topics are centered in the area of wireless LAN, video compression, QoS video transmission and error-resilient protocol architectures. He has published over 100 papers in international Journals and Conferences. He has served in the organization of International Conferences as Chair, Technical Program Chair and Technical Program Committee member. He was the Chair of the IFIP 6.8 Working Group.

A. Garrido received the degree in Physics (Electronics and Computer Science) and the Ph.D. degrees from the University of Granada, Spain, in 1986 and University of Valencia, Spain, in 1991, respectively. In 1986, he joined the Department of Computer Engineering at the University of Castilla-La Mancha, where he is currently a Full Professor of Computer Architecture and Technology. His research interests include high-performance networks, telemedicine, video compression, and video transmission. He has published over 40 papers in international journals and he has led several research projects. He collaborates with the Spanish Ministry of Education in the National plan for quality evaluation of Spanish Universities from the year 2000. He has been dean of the School of Computer Engineering from 2000 to 2008.