Fast Mode Decision Algorithm for H.264/AVC-to-SVC Transcoding with Temporal Scalability

Rosario Garrido-Cantos[1], Jan De Cock[2], Sebastiaan Van Leuven[2], Pedro Cuenca[1], Antonio Garrido[1], and Rik Van de Walle[2]

[1] Albacete Research Institute of Informatics, University of Castilla-La Mancha, Albacete, Spain
{charo,pcuenca,antonio}@dsi.uclm.es
[2] Department of Electronics and Information Systems - Multimedia Lab,
Ghent University - IBBT, Ghent, Belgium
{jan.decock,sebastiaan.vanleuven,rik.vandewalle}@ugent.be

Abstract. Scalable Video Coding (SVC) uses a notion of layers within the encoded bitstream for providing temporal, spatial and quality scalability, separately or combined. By truncating layers the bitstream can be adapted to devices with different characteristics and to varying network constraints. Since the majority of the existing video content is encoded using H.264/AVC without scalability, they cannot benefit from these scalability tools, so a transcoding process should be applied to provide scalability to this existing encoded content. In this paper, an algorithm based on Machine Learning techniques for temporal scalability transcoding from H.264/AVC to SVC focusing on mode decision task is discussed. The results show that when our technique is applied, the complexity is reduced by 82% while maintaining coding efficiency.

Keywords: Scalable Video Coding (SVC), H.264/AVC, Transcoding, Temporal Scalability, Machine Learning.

1 Introduction

The users' demand for multimedia content such as video streaming in digital video services has grown spectacularly in the last years. These video streams, generally, are encoded to reduce the necessary storage and the network bandwidth for transmission. In 2007, the scalable extension of the H.264/AVC [1][2] video coding standard was finalized. *Scalable Video Coding* (SVC) [3] provides temporal, spatial, quality scalability or a combination of these. An 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, resolution 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 digital video streams are still created in a single-layer format (such as H.264/AVC video streams) so they cannot benefit from this scalability. This

K. Schoeffmann et al. (Eds.): MMM 2012, LNCS 7131, pp. 585–596, 2012.

fact leads to the need for developing alternative techniques to enable video adaptation between non-scalable and scalable bitstreams. In this framework, transcoding approaches are one of the solutions used to adapt a video stream by reducing the temporal resolution, lowering the spatial resolution, decreasing the visual quality, or changing the coding format.

In this paper, an efficient video transcoding [4] technique is proposed for transforming H.264/AVC bitstreams to SVC bitstreams providing temporal scalability. The ultimate goal is to perform the required adaptation process faster than the straightforward concatenation of decoder and encoder while maintaining its coding efficiency. In the H.264/AVC standard and its SVC extension, inter prediction is carried out by means of variable block size motion estimation, which is able to eliminate the temporal redundancy between two or more adjacent frames. This approach supports motion compensation block sizes ranging between 16x16, 16x8, 8x16 and 8x8, where each of the sub-divided regions is a *Macroblock* (MB) partition. If the 8x8 mode is chosen, each of the four 8x8 block partitions within the MB may be further split in 4 ways: 8x8, 8x4, 4x8 or 4x4, which are known as sub-MB partitions. In the proposed approach, the reduction in complexity is obtained by reusing as much information as possible from the original bitstream, such as H.264/AVC mode decision, residual, etc. to reduce the encoding SVC time focusing on the mode decision process. This time saving is achieved by narrowing the possible modes where the standard can choose by using a decision tree built using *Machine Learning* (ML) tools. This technique is applied to different sequences using varying GOP sizes in Baseline Profile.

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. Sect. 3 introduces briefly the temporal scalability technique in SVC. In Sect. 4, our proposal is described. In Sect. 5 the results of applying our approach are presented and, finally, in Sect. 6 conclusions are shown.

2 Related Work

In the literature different proposals exist for using *Machine Learning* for transcoding. Some examples are [5][6].

In the framework of H.264/AVC-to-SVC video transcoding, in the last few years, different techniques have been proposed. They can be classified into three different types of scalability.

For quality-SNR scalability, in 2006 *Shen at al. proposed* a technique for transcoding from hierarchically-encoded H.264/AVC to Fine-Grain Scalability (FGS) streams [7]. In 2009, *De Cock et al.* presented different open-loop architectures for transcoding from a single-layer H.264/AVC bitstream to SNR-scalable SVC streams with *Coarse-Grain Scalability* (CGS) layers [8]. In 2010 and 2011, they proposed simple closed-loop architectures that reduce the time of the mode decision [9][10].

Regarding spatial scalability, in 2009 a proposal was presented by *Sachdeva et al.* in [11]. The idea consists of an information single layer to SVC multiple-layer for adding spatial scalability to all existing non-scalable H.264/AVC video streams. The algorithm reuses available data by an efficient downscaling of video information for different layers.

Finally, for temporal scalability, in 2008 a transcoding method from an H.264/AVC P-picture-based bitstream to an SVC bitstream was presented in [12] 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 [13]. 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. The same year we 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 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 [14]. Later, in 2011, the previous algorithm was adjusted for Baseline Profile and P frames [15]. At this point, we emphasize that the present work is another step in the framework of H.264/AVC to SVC video transcoders. Our previous approaches [14][15] focused only on the motion estimation process, where the idea consists in reducing the search area dynamically based on the incoming H264/AVC motion vectors. On the contrary, in this work, while the motion estimation is kept untouched, the approach is extended to a MB mode decision algorithm which is explained in next section. As future work, we can try to combine both approaches together.

3 Temporal Scalability in SVC

Since our proposal focuses on temporal scalability, a brief explanation about this type of scalability is given in this section. For a comprehensive overview of the whole scalable extension of H.264/AVC, the reader is referred to [3].

As it was said in the introduction, in a sequence with temporal scalability, the bitstream is encoded in layers. The base layer (with an identifier equal to 0) represents the lowest frame rate while the temporal enhancement layers (with identifiers that increase by 1 in every layer) increase the available frame rate. By removing temporal layers, the frame rate can be adapted dynamically.

Fig. 1 shows a sequence with a *Group of Pictures* (GOP) of 8 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.

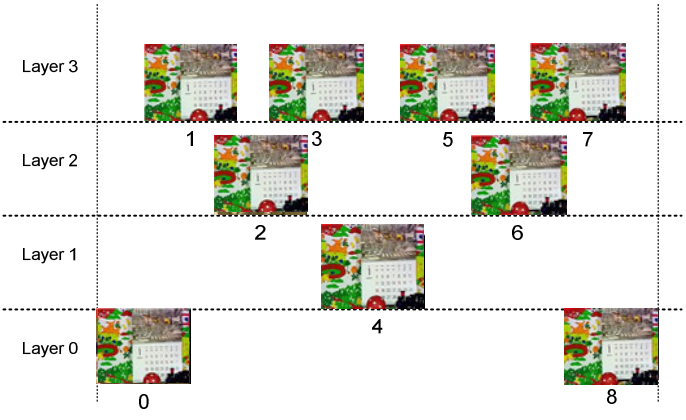


Fig. 1. Distribution per temporal layer of the frames within a GOP = 8

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 were provided that make it possible to mark any picture as a reference picture, so that it can be used for motion-compensated prediction of the following pictures. In this way, to achieve temporal scalability, SVC links its reference and predicted frames using hierarchical prediction structures [16] which define the temporal layering of the final structure. 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. There are different structures for enabling temporal scalability, but the one used by default in the *Joint Scalable Video Model* (JSVM) reference encoder software [17] is based on hierarchical pictures with a dyadic structure where the number of temporal layers is thus equal to $1 + \log_2[GOP\ size]$.

Temporal scalability based on P pictures was introduced in [18]. This technique provides lower latency and is particularly useful for multimedia communications like mobile 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.

4 Proposed H.264/AVC-to-SVC Video Transcoder

4.1 Motivation

One of the computationally most intensive tasks involved in the SVC encoding process is the MB mode decision. Therefore, this is one of the more suitable parts of the proposed H.264/AVC-to-SVC transcoder to be accelerated. H.264/AVC and its extension SVC support both intra prediction and inter prediction in P or B frames. Intra prediction only requires data from the current picture, while inter prediction uses data from pictures that have been previously coded and transmitted (reference pictures) and is used for eliminating temporal redundancy in P and B frames. As it has

been depicted before, SVC supports different MB and sub-MB partitioning modes for inter prediction as shown in Fig. 2. Moreover, SVC also allows intra predicted modes, and a skipped mode in inter frames for referring to the 16x16 mode where no motion and residual information is encoded.

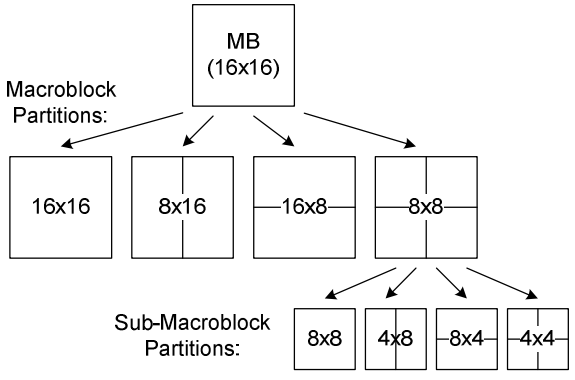
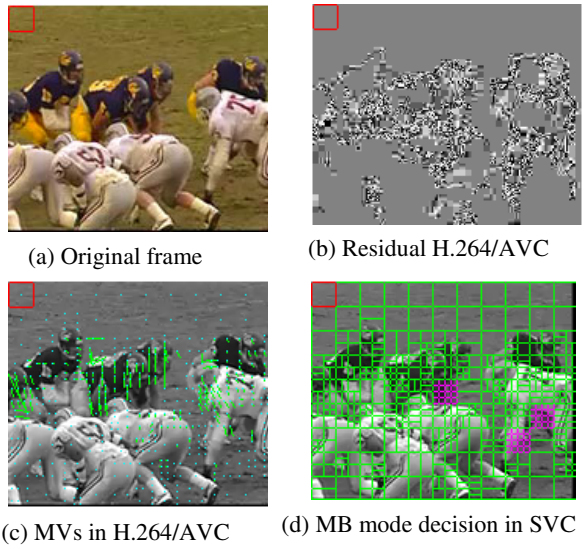


Fig. 2. Macroblock and sub-macroblock partitions for inter prediction

Since for choosing the best partitioning, every block size is checked by the SVC encoder (as part of the proposed transcoder), a way to reduce the time spent by this mode decision process is trying to narrow the set of possible MB partitions. This paper proposes an algorithm which makes use of some information gathered in the H.264/AVC decoding algorithm to reduce the MB partitions to be checked in the SVC encoder. By using a data mining procedure, an algorithm which implements a decision tree is proposed. This decision tree is generated by means of machine learning tools.

Although the prediction structure (and as result, the frames used as a reference) of H.264/AVC without temporal scalability (in this case using the well-known IPPP pattern) and SVC are not the same, some data extracted from the H.264/AVC decoding algorithm can still be reused in the transcoder. This information can help us to find out the best partitioning structure without the needed of checking all the MB partitions. This correlated information extracted from the H.264/AVC decoding algorithm is depicted in Fig. 3. In this figure, the correlation between the residual and MV length calculated in H.264/AVC with respect to the MB coded partition done in SVC are shown.

In this case, after an extensive analysis, we observed that stationary areas or objects with slow motion are often coded in MBs without sub-blocks (such as 16x16, 16x8 or 8x16) or even as Skipped where the MB contains no residual data. On the other hand, the regions with sudden changes (scene, light, an object that appears) are coded using inter modes with smaller MB mode partitions (such as 4x8, 8x4, 4x4) or even using the Intra mode. Moreover, we also found a high correlation between the length of the MVs calculated by H.264/AVC and the final MB mode decision where long MVs suggest more complicated MB partitions such as 4x4, while shorter MVs lead to simpler MB partitions. These relationships can be observed in Fig. 3 as well. Taking



(a) Original frame (b) Residual H.264/AVC

(c) MVs in H.264/AVC (d) MB mode decision in SVC

Fig. 3. Exploiting the correlation using Machine Learning

into account these observations, the information that needs to be extracted from the H.264/AVC decoding process is:

- Residual: The residual data of every block of 4x4 pixels is used by the decoder to reconstruct the MB, so this information will be available in the decoding process. For our purpose, only the residual data of the luma component is extracted.
- Motion vectors: This information is available as well in the decoding process. The motion vectors of each MB are extracted. Note that each MB in H.264/AVC can have more than one pair of motion vectors since each MB can be further divided in smaller partitions.
- Mode decision of H.264/AVC: The MB partitioning of each MB in H.264/AVC is related to the residual and the motion vectors and can give us valuable information.

4.2 Generating the Decision Tree

Data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. This information can be converted into knowledge. 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. For this proposal, we used ML tools for converting the relationships between data extracted from the H.264/AVC decoding process and the MB mode partitioning of SVC into a decision tree. By using this decision tree, the possible modes that can be chosen by the encoder are narrowed down.

This decision tree was built using the WEKA software [19]. WEKA is a collection of machine learning algorithms for data mining tasks and also contains tools for data

pre-processing, classification, regression, clustering, association rules, and visualization. For every MB, the extracted information is used to generate the decision tree (used to decide the MB partitioning later). Some operations and statistics are calculated for this data such as the length of the motion vectors, the variance of means of the residual of 4x4 blocks or the mean of variances of the residual of these blocks.

The information enumerated in Section 4.1 together with the SVC encoder mode decision was introduced and then, an ML classifier was run. In this case, the well-known RIPPER algorithm [20] was used. The process for building the decision tree for H.264/AVC-to-SVC transcoding is shown in Fig. 4. The obtained binary decision tree has three decision levels:

- First level: Discriminates between LOW {SKIP, 16x16, 16x8, 8x16} and HIGH COMPLEXITY {INTRA, 8x8, 8x4, 4x8, 4x4} modes.
- Second level: Inside the LOW COMPLEXITY bin, a decision between {SKIP, 16x16} or {16x8, 8x16} is made.
- Third level: Inside the HIGH COMPLEXITY bin, a decision between {8x8, 8x4, 4x8} or {4x4, INTRA} is made.

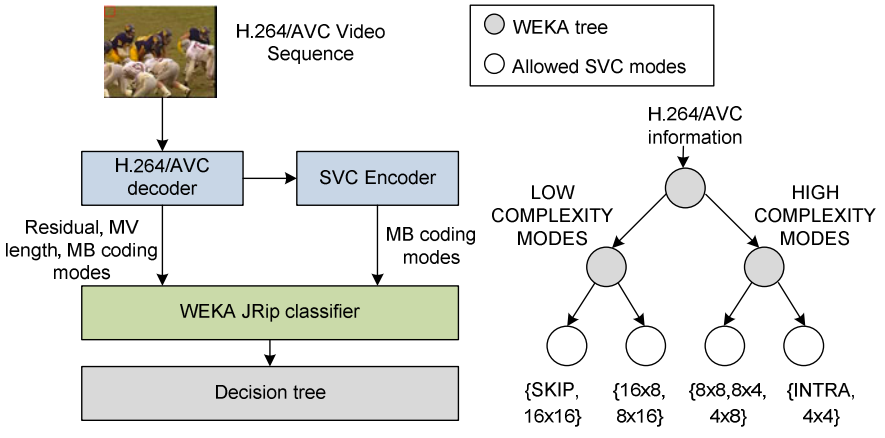


Fig. 4. Building the decision tree

This tree was generated with the information available after the decoding process and does not focus on the final MB partition, but reduces the set of MB modes that can be chosen by SVC encoder. This is represented in Fig. 4 where the white circles represent the set of MB partitions the SVC encoder can choose from. The ML process gives us a decision tree that classifies correctly in about 87% of the cases in the 1st level, 80% in the 2nd level and 93% in the 3rd level. For all this cases, training with *Football* and testing with the rest of sequences of the performance evaluation (see Section 5).

This decision tree is 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 for each QP. The solution is to develop a single decision tree for a specific QP and adjust the mean and the variance threshold used by the trees based on the QP.

5 Implementation Results

In this section, results from the implementation of the proposal described in the previous section are shown. Test sequences with varying characteristics were used, namely *Foreman, Harbour, Mobile, City, Soccer* and *Hall* in CIF resolution (30 Hz) and QCIF resolution (15 Hz). These sequences were encoded using the H.264/AVC *Joint Model* (JM) reference software [21], version 16.2, with an IPPP pattern and a fixed QP = 28 in a trade-off between quality and bitrate. Then, for the reference results, the encoded bitstreams are decoded and re-encoded using the JSVM software [17], version 9.19.3 [17] with temporal scalability, Baseline Profile and different values of QP (28, 32, 36, 40).

For the results of our proposal, encoded bitstreams in H.264/AVC are transcoded using the technique described in the previous section. This technique was applied to the enhancement temporal layers because, as it was shown in [15], the two enhancement temporal layers with the highest identifier are where most encoding time is spent. In these results, the *Football* sequence has been excluded from the evaluation set since it was used as a training sequence for generating the decision tree.

In Table 1, 2 and 3 the results for ΔPSNR, ΔBitrate and Time Saving are shown when our technique is applied compared to the reference transcoder. ΔPSNR and ΔBitrate are calculated according to the *Bjøntegaard-Delta* metric [22].

Time Savings are calculated for the full sequence ('Full Seq.') and for the temporal layers where the technique is applied to ('Partial'). To evaluate it, (1) is calculated where T_{ref} denotes the coding time used by the SVC reference software encoder and T_{pro} is the time spent by the proposed algorithm.

$$T_{saving}(\%) = 100 \cdot (T_{ref} - T_{prop})/T_{ref} \quad (1)$$

Table 1. RD performance and time savings of the approach for GOP = 2 and different resolutions

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 2								
	QCIF (15 Hz)				***CIF (30 Hz)***			
Sequence	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
			Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	0.042	-0.05	57.96	85.38	0.055	-0.08	58.94	86.64
City	0.026	0.92	57.16	84.16	0.055	0.25	58.24	85.61
Foreman	0.077	1.21	56.20	82.70	-0.059	1.51	58.12	85.46
Soccer	0.036	1.45	54.34	79.86	0.021	1.28	56.28	82.85
Harbour	0.022	-0.13	52.91	77.95	0.047	-0.35	56.12	80.58
Mobile	0.033	-0.15	52.28	76.93	0.080	-1.10	54.51	80.09
Average	***0.039***	***0.54***	***55.14***	***81.16***	***0.033***	***0.25***	***57.03***	***83.54***

ΔPSNR: Difference in quality (negative means quality loss); ΔBitrate: Bitrate increase; Time Saving: complexity reduction.

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 65% of reduction of computational complexity in the full sequence and 82% in the specific layers is achieved. The resulting Rate-Distortion (RD) curves for the CIF SVC bitstream with GOP = 4 is shown in Fig. 5 where it can be seen that our proposal for transcoding is able to approach the RD of the reference transcoded (re-encoded) without any significant coding efficiency loss. Due to the space restriction, only one RD plot can be shown; the rest of the figures are similar.

Table 2. RD performance and time savings of the approach for GOP = 4 and different resolutions

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 4								
	QCIF (15 Hz)				***CIF (30 Hz)***			
Sequence	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
			Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	0.219	0.04	74.58	85.80	0.328	-0.45	74.69	86.45
City	0.064	1.93	75.69	86.04	0.200	0.66	76.30	86.96
Foreman	0.251	2.34	72.68	83.55	-0.112	3.01	74.63	85.65
Soccer	0.043	2.24	72.11	81.83	0.021	2.37	72.35	83.05
Harbour	0.107	-0.68	68.30	78.88	0.175	-1.22	71.75	81.57
Mobile	0.142	0.15	65.37	76.51	0.229	-1.69	69.83	80.37
Average	***0.138***	***1.00***	***71.46***	***82.10***	***0.140***	***0.45***	***73.26***	***84.01***

Table 3. RD performance and time savings of the approach for GOP = 8 and different resolutions

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 8								
	QCIF (15 Hz)				***CIF (30 Hz)***			
Sequence	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
			Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	0.158	0.37	70.59	86.28	0.025	0.47	70.69	86.83
City	-0.008	2.67	70.16	85.70	0.175	1.32	70.10	86.16
Foreman	0.210	3.22	66.89	82.89	-0.001	3.58	69.96	85.91
Soccer	0.074	2.61	65.19	80.63	-0.001	2.99	68.07	83.55
Harbour	0.048	0.15	64.60	79.54	0.072	-0.18	65.54	80.60
Mobile	0.031	0.87	64.82	79.36	0.233	-0.84	65.81	81.10
Average	***0.086***	***1.65***	***67.04***	***82.40***	***0.084***	***1.22***	***68.36***	***84.02***

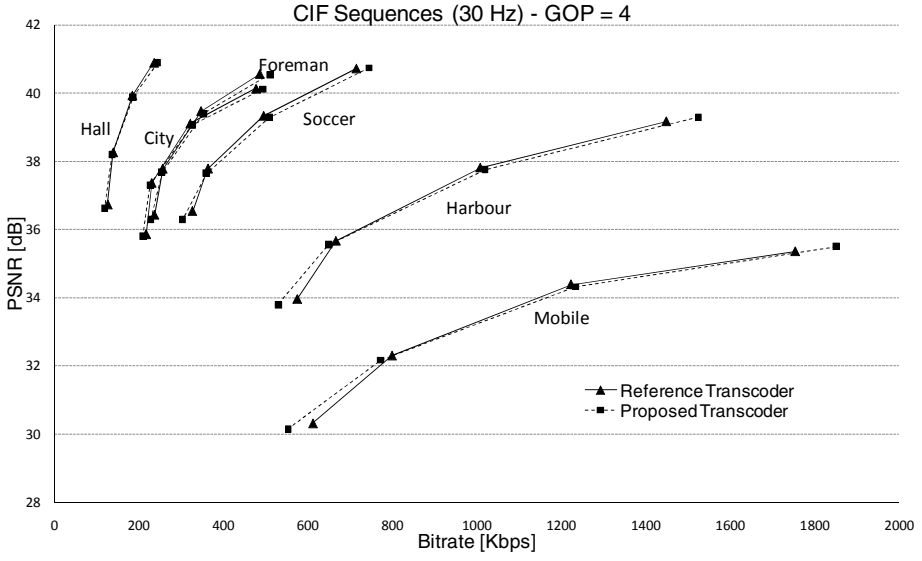


Fig. 5. Rate-distortion performance of CIF sequences with GOP = 4

Finally, our technique is capable of outperforming earlier solutions such as in [12][13][15]. In contrast to [12], we show that our proposal can be successfully applied to a wide range of test sequences with varying motion characteristics and resolutions. A comparison with these proposals is shown in Table 4. This comparison is done with the values available in the papers (PSNR and Time Saving for Foreman CIF with GOP = 2 and an average of PSNR and Time Saving for different sequences in QCIF and CIF resolutions and GOP = 8). Regarding ΔBitrate, there is not numerical information in [12] and [13] and comparing to [15] the ΔBitrate is similar.

Table 4. Comparison between different proposals

Comparison with other proposals						
	GOP = 2 CIF - Foreman		***GOP = 8 QCIF***		***GOP = 8 CIF***	
Proposal	**ΔPSNR (dB)**	**TS (%)**	**ΔPSNR (dB)**	**TS (%)**	**ΔPSNR (dB)**	**TS (%)**
Dziri et al. [12]	-0.50	47.00	--	--	--	--
Al-Muscati et al. [13]	-0.50	37.00	-0.200	55.20	--	62.10
Garrido-Cantos et al. [15]	-0.01	41.79	-0.027	51.90	-0.040	48.83
Our technique	**-0.06**	**58.12**	**0.086**	**67.04**	**0.084**	**68.36**

ΔPSNR: Difference in quality (negative means quality loss); TS: complexity reduction.

6 Conclusions

In this paper, a proposal based on Machine Learning tools for transcoding H.264/AVC bitstreams to SVC streams with temporal scalability 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 macroblock mode decision process is reduced. The experimental results show that it is capable to reduce the coding complexity by around 82% where it is applied while maintaining the coding efficiency.

Acknowledgments. This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under the grants CSD2006-00046, TIN-2009-14475-C04, and it was also partly supported by JCCM funds under grants PEII09-0037-2328 and PII2I09-0045-9916.

References

1. ITU-T and ISO/IEC JTC 1: Advanced Video Coding for Generic Audiovisual Services. In: ITU-T Rec. H.264/AVC and ISO/IEC 14496-10 (including SVC extension) (March 2009)
2. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Transaction on Circuits and System for Video Technology 13(7), 560–576 (2003)
3. Schwarz, H., Marpe, D., Wiegand, T.: 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 (2007)
4. Vetro, A., Christopoulos, C., Sun, H.: Video Transcoding Architectures and Techniques: an Overview. IEEE Signal Processing Magazine, 18–29 (2003)
5. Martinez, J.L., Fernandez-Escribano, G., Kalva, H., Fernando, W.A.C., Fernando, Cuenca, P.: Wyner-Ziv to H.264 Video Transcoder for Low Cost Video Communications. IEEE Transaction on Consumer Electronics 55(3), 1453–1461 (2009)
6. Fernandez-Escribano, G., Bialkowski, J., Gamez, J.A., Kalva, H., Cuenca, P., Orozco-Barbosa, L., Kaup, A.: Low-Complexity Heterogeneous Video Transcoding Using Data Mining. IEEE Transactions on Multimedia 10(2), 286–299 (2008)
7. Shen, H., Sun, X.S., Wu, F., Li, H., Li, S.: Transcoding to FGS Streams from H.264/AVC Hierarchical B-Pictures. In: IEEE Int. Conf. Image Processing, Atlanta (2006)
8. De Cock, J., Notebaert, S., Lambert, P., Van de Walle, R.: Architectures of Fast Transcoding of H.264/AVC to Quality-Scalable SVC Streams. IEEE Transaction on Multimedia 11(7), 1209–1224 (2009)
9. Van Wallendael, G., Van Leuven, S., Garrido-Cantos, R., De Cock, J., Martinez, J.L., Lambert, P., Cuenca, P., Van de Walle, R.: Fast H.264/AVC-to-SVC transcoding in a mobile television environment. In: Proceedings of Mobile Multimedia Communications Conference, 6th International ICST, Lisbon (2010)
10. Van Leuven, S., De Cock, J., Van Wallendael, G., Van de Walle, R., Garrido-Cantos, R., Martinez, J.L., Cuenca, P.: Combining Open- and Closed-loop Architectures for H.264/AVC-to-SVC Transcoding. In: 18th IEEE International Conference on Image Processing (in press)
11. Sachdeva, R., Johar, S., Piccinelli, E.: Adding SVC Spatial Scalability to Existing H.264/AVC Video. In: 8th IEEE/ACIS International Conference on Computer and Information Science, Shangai (2009)

12. Dziri, A., Diallo, A., Kieffer, M., Duhamel, P.: P-Picture Based H.264 AVC to H.264 SVC Temporal Transcoding. In: International Wireless Communications and Mobile Computing Conference (2008)
13. Al-Muscati, H., Labeau, F.: Temporal Transcoding of H.264/AVC Video to the Scalable Format. In: 2nd Int. Conf. on Image Processing Theory Tools and Applications, Paris (2010)
14. Garrido-Cantos, R., De Cock, J., Martínez, J.L., Van Leuven, S., Cuenca, P., Garrido, A., Van de Walle, R.: Video Adaptation for Mobile Digital Television. In: IFIP Wireless and Mobile Networking Conference, Budapest, Hungary (2010)
15. Garrido-Cantos, R., De Cock, J., Martínez, J.L., Van Leuven, S., Cuenca, P.: Motion-Based Temporal Transcoding from H.264/AVC-to-SVC in Baseline Profile. IEEE Transactions on Consumer Electronics 57(1) (February 2011)
16. Schwarz, H., Marpe, D., Wiegand, T.: Analysis of Hierarchical B pictures and MCTF. In: IEEE Int. Conf. ICME and Expo, Toronto (2006)
17. Joint Video Team (JSVM) reference software, http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm
18. Wenger, S.: Temporal scalability using P-pictures for low-latency applications. In: IEEE Second Workshop on Multimedia Signal Processing, Redondo Beach, CA, USA, pp. 559–564 (December 1998)
19. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
20. Cohen, W.W.: Fast Effective Rule Induction. In: 20th International Conference on Machine Learning, pp. 115–123 (1995)
21. Joint Model JM reference software, http://iphome.hhi.de/suehring/tml/download/
22. Sullivan, G., Bjøntegaard, G.: 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)