View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Ghent University Academic Bibliography

UNIVERSITEIT GENT

**biblio.ugent.be**

The UGent Institutional Repository is the electronic archiving and dissemination platform for all UGent research publications. Ghent University has implemented a mandate stipulating that all academic publications of UGent researchers should be deposited and archived in this repository. Except for items where current copyright restrictions apply, these papers are available in Open Access.

This item is the archived peer-reviewed author-version of:

Scalable Video Transcoding for Mobile Communications

Rosario Garrido-Cantos, Jan De Cock, Jose Luis Martinez, Sebastiaan Van Leuven, Pedro Cuenca, and Antonio Garrido

In: Springer Telecommunication Systems, 55 (2), 173-184, 2014.

**To refer to or to cite this work, please use the citation to the published version:**

**Garrido-Cantos, R., De Cock, J., Martinez, J. L., Van Leuven, S., Cuenca, P., and Garrido, A. (2014). Scalable Video Transcoding for Mobile Communications. *Springer Telecommunication Systems 55(2)* 173-184.**

# Scalable Video Transcoding for Mobile Communications

R. Garrido-Cantos[1], J. De Cock[2], J.L. Martínez[1], S. Van Leuven[2], P, Cuenca[1], and A. Garrido[1]

[1]*Albacete Research Institute of Informatics, University of Castilla-La Mancha*
*Campus Universitario s/n, 02071 Albacete, Spain*
{charo, joseluismm, pcuenca, antonio}@dsi.uclm.es

[2]*Department of Electronics and Information Systems, Multimedia Lab, Ghent University-IBBT*
*Gaston Crommenlaan 8, bus 201, B-9050 Ledeberg, Ghent, Belgium*
{jan.decock, sebastiaan.vanleuven}@ugent.be

**Abstract.** Mobile multimedia contents have been introduced in the market and their demand is growing every day due to the increasing number of mobile devices and the possibility to watch them at any moment in any place. These multimedia contents are delivered over different networks that are visualized in mobile terminals with heterogeneous characteristics. To ensure a continuous high quality it is desirable that this multimedia content can be adapted on-the-fly to the transmission constraints and the characteristics of the mobile devices. In general, video contents are compressed to save storage capacity and to reduce the bandwidth required for its transmission. Therefore, if these compressed video streams were compressed using scalable video coding schemes, they would be able to adapt to those heterogeneous networks and a wide range of terminals. Since the majority of the multimedia contents are compressed using H.264/AVC, they cannot benefit from that scalability. This paper proposes a technique to convert an H.264/AVC bitstream without scalability to a scalable bitstream with temporal scalability as part of a scalable video transcoder for mobile communications. The results show that when our technique is applied, the complexity is reduced by 98% while maintaining coding efficiency.

*Keywords: Mobile Video, Video Adaptation, H.264/AVC, Scalable Video Coding (SVC), Temporal Scalability;*

## 1. Introduction

Consumers' demand for content on the move is growing day after day. They want to access the content they like, when and where they want. One of the most requested services are those that allow receiving and watching multimedia contents (movies, TV programs, live retransmissions, etc.) on mobile devices.

There are different alternatives to transmit these contents from broadcasters to the users. One of them is the Mobile Internet Protocol Television (IPTV) [21] which delivers digital multimedia contents by Internet protocols, so users who have any kind of IP devices are able to watch various multimedia services including television on mobile terminals on the move. 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 [7]. All of them are extensions of other existing networks for fixed services and introduce improvements for trying to overcome the difficulties for transmitting in mobile environments (fluctuating bandwidths, indoor reception, etc.). Even so, issues with the dynamic environment, the vulnerability of the links of transmission or bandwidth limitation are present.

For delivering video over these networks ensuring continuous high quality image, it is important that the contents can be adapted to the receivers and the varying networks. 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 adaptation for mobile devices plays a crucial role.

In general, the contents transmitted over these networks are compressed to make easier their storage and to reduce the bandwidth necessary for the transmission, so it is desirable that these compressed contents can adapt to these different terminals and networks [5][13][16]. This is possible using Scalable Video Coding (SVC) [12].

SVC provides different types of scalability such as temporal, spatial, quality or a combination allowing this adaptation on-the-fly. SVC was standardized in 2007 and is based on H.264/AVC [12]. The SVC video stream is divided into layers, one base layer which represents the lowest frame rate, the lowest spatial and the lowest quality resolutions and one or more enhancement layers which increase frame rate, add more spatial resolutions or more quality. By removing certain layers from the original bitstream, it is possible to adapt to the communication channel bandwidth and/or user device capabilities at every moment.

Nowadays, most of the video content is still created in H.264/AVC without any type of scalability. To transform these existing video streams in H.264/AVC to SVC, which can be adapted to different network characteristics and user terminals, a video transcoder is proposed. Heterogeneous transcoding [26] is a technique for adaptation or conversion of one encoding format to another. 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. This transcoding step can be applied at the broadcaster side as shown in Fig. 1.

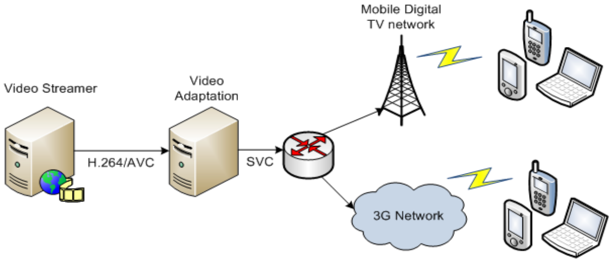


Fig. 1. Example of an SVC transcoder for mobile environments

The goal is to perform the required adaptation process faster than the concatenation of decoder and encoder. In particular, in this paper we propose a low complexity transcoder for transforming H.264/AVC bitstreams in Baseline Profile (P-picture based) without temporal scalability (frame rate variability) into an SVC bitstream with temporal scalability. That transcoding technique is capable to reduce coding complexity around 98% while maintaining coding efficiency.

Our approach is based on reusing information such as residual data, motion vectors (MV), mode decision, etc. in the H.264/AVC decoding stage in order to reduce the mode decision and motion estimation tasks in SVC. The mode decision task is accelerated by building a decision tree using data mining techniques and then, uses it to narrow the possible mode decision. This idea is based on the high correlation between this information and the final SVC mode decision. The motion estimation task is accelerated by reducing the search area of the SVC encoder by building a new one using the information of the MV collected from the H.264/AVC bitstream. This is possible because those MVs represent approximately the amount of movement of the scene.

The remainder of this paper is organized as follows. In Section 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 Section 4 our approach is depicted. In Section 5 the implementation results are shown. Finally, in Section 6 conclusions are presented.

## 2. Related Work

In the last few years, different techniques for transcoding from H.264/AVC-to-SVC have been proposed. Most of the proposals are related to quality-SNR scalability, although there are few related to spatial and temporal 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 [18]. Although it was the first work in this type of transcoding, it does not have much relevance since this technique for providing quality-SNR scalability was removed from the following versions of the standard due to its high computational complexity. 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 [4]. In 2010, Van Wallendael et al. proposed a simple closed-loop architecture that reduces the time of the mode decision process by analyzing the mode information from the input H.264/AVC video stream and using it to build a fast mode decision model [25]. Then, in 2011, Van Leuven et al. proposed two techniques to improve the previous proposals [23][24].

Regarding spatial scalability, in 2009 a proposal was presented by Sachdeva et al. in [17]. 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 [6] 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 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 [8]. Later, in 2011, the previous algorithm was adjusted for the Baseline Profile and P frames [9]. In the same year, we presented another work [10] focusing in accelerating the mode decision algorithm, while our previous approaches focused only on motion estimation process. The present work is another straightforward step in the framework of H.264/AVC to SVC video transcoders where the approaches presented in [9] and [10] has been combined and adjusted to work together.

# 3. Technical Background

## 3.1 Scalable Video Coding

Scalable Video Coding 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 into 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:

*1) 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.

*2) 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.

*3) 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.

Our proposal is based in provides temporal scalability to a bitstream, so we are going to explain this with more detail below.

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. Fig. 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.

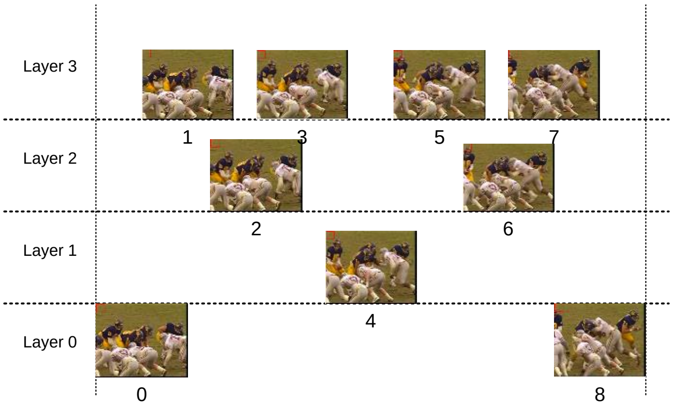


Fig. 2. Sequence with temporal scalability. Distribution of the eight first frames per every layer

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 [20] 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.

There are different structures for enabling temporal scalability, but the one used by default in the Joint Scalable Video Model (JSVM) reference encoder software [15] 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 [20][27]. 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 qualities.

For a comprehensive overview of the scalable extension of H.264/AVC, the reader is referred to [20].

## 3.2 Motion Estimation Process

The motion estimation process consists in finding a region in a reference frame that matches as much as possible to the current macroblock (MB). In order to find this region, search area situated in the reference frame is defined. That search area is centered on the current macroblock position and the region within the search area that minimizes a matching criterion is chosen. For elimination the temporal redundancy, motion vectors between every MB or sub-MB and that block which generates the most appropriate match inside the search area of the reference frame are calculated. The process is illustrated in Fig. 3.

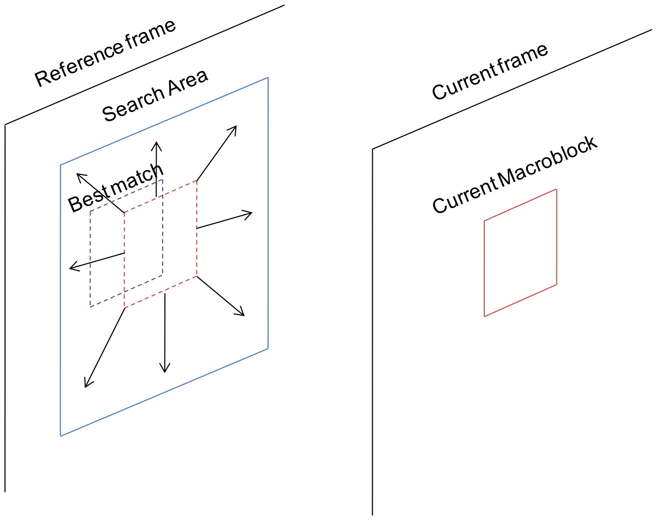


Fig. 3. Motion Estimation Process

### 3.3 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 16x16, 16x8, 8x16 to 8x8; where each of the sub-divided regions is an 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. 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. 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.4.

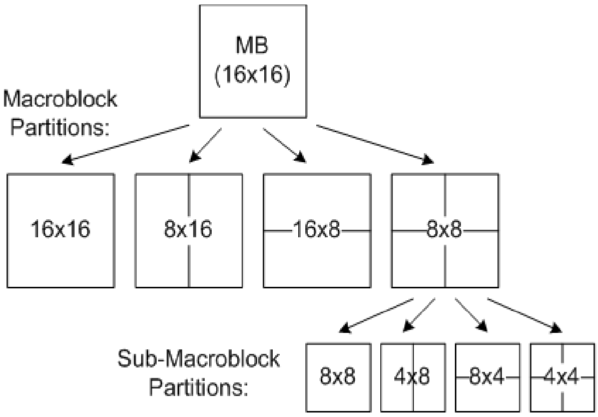


Fig. 4. Macroblock and sub-macroblock partitions for inter prediction

# 4. Proposed Video Transcoding

One of the computationally most intensive tasks involved in the SVC encoding process is the inter-prediction process. This process involves the mode decision and the motion estimation tasks described previously. The key idea behind the proposed transcoder is to accelerate those tasks using information collected in the H.264/AVC decoding stage. On the one hand, MB mode decision is accelerated using a decision tree that has been obtained using Machine Learning tools and, on the other hand motion estimation process was accelerated by reducing the search area. Both proposals are combined and adjusted to work together.

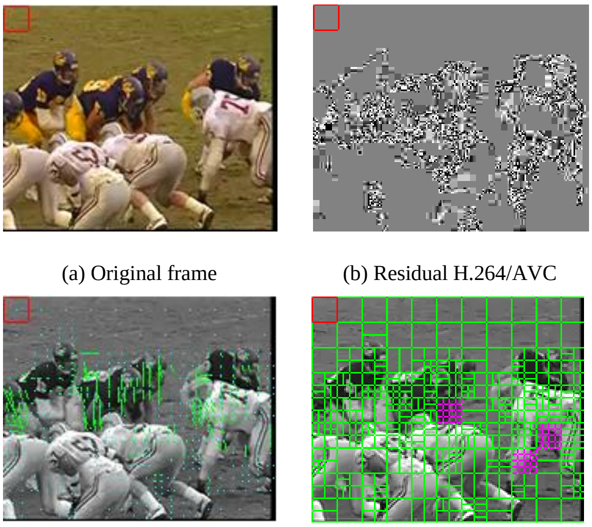
In the next subsections we will describe these algorithms.

## 4. 1 Fast Mode Decision Algorithm

As mentioned previously, the SVC encoder part of the H.264/AVC-to-SVC transcoder takes a large amount of time for searching exhaustively all inter and intra modes to select the best for each macroblock.

The main goal of this proposal was to reduce the time spent by this mode decision process, trying to narrow the set of macroblock partitions to be checked by the encoder by using a decision tree generated by data mining techniques.

Although the prediction structure (and, as a result, the frames used as a reference) of H.264/AVC without temporal scalability (in this case using an IPPP pattern) and SVC are not the same, some data generated by H.264/AVC and transmitted into the encoded bitstream can help us to find out the best partitioning structure. For example, in Fig. 5, 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.



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

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

Fig. 5. Exploiting the correlation using Machine Learning

Taking into account these observations, the information that need to be extracted from the H.264/AVC decoder process will be:

*1) Residual*: The amount of residual of every block of 4x4 pixels is used by the decoder to reconstruct the decoded macroblock, so this information will be available in the decoding process. For our purpose, only the residual data of the luma component was extracted.

*2) Motion vectors:* This information is available as well in the decoding process. The motion vectors of each MB were extracted.

Mode decision of H.264/AVC: The macroblock partitioning of each MB in H.264/AVC is related to the residual and the motion vectors and can give us valuable information.

We emphasize that this information is gathered inside the transcoder, more in particular, in the H.264/AVC decoding part, and that it only has to be passed to the second half of the transcoder (SVC encoder part).

### *4.1.1 Generating the decision tree*

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, a decision tree with three levels of decision is presented. This decision tree narrows the mode decisions that can be chosen by the standard.

To build the decision tree we used the WEKA software [11]. 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 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:

*1) Residual of the whole macroblock:* The residual of all the 4x4 blocks of pixels (res4x4) within the MB are added.

*2) Length of the average of the motion vectors of a macroblock:* First of all, the mean of each component of all the MVs of the H.264/AVC MB and sub-MB is calculated. This MV is the motion vector of the MB that we will use. Then, the length of the resulting MV is calculated.

*3) Mean of variances of the residual of 4x4 blocks within a macroblock:* For every block of 4x4 pixels, the variance of the its residuals is calculated. Then, the mean of the variances resulting of this process is done.

*4) Variance of means of the residual of 4x4 blocks within a macroblock:* For every block of 4x4 pixels, the mean of its residuals is calculated. Then, the variance of these means is done.

The information enumerated together with the SVC encoder mode decision were introduced and then, an ML classifier was run. In this case, the well-known RIPPER algorithm [3] was used. The process for building the decision tree for H.264/AVC-to-SVC transcoding is shown in Fig. 6. The training file was generated using the sequence Football and only taking into account the frames within the enhancement temporal layer.

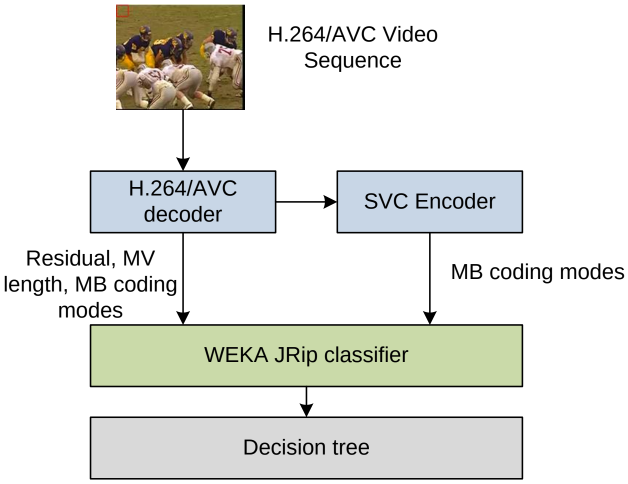


Fig. 6. Process for building the decision tree for H.264/AVC-to-SVC transcoding

The binary decision tree obtained has three decision levels:

*1) 1st level:* Discriminates between LOW {SKIP, 16x16, 16x8, 8x16} and HIGH COMPLEXITIY {INTRA, 8x8, 8x4, 4x8, 4x4} modes.

*2) 2nd level:* Inside the LOW COMPLEXITY bin, a decision between {SKIP, 16x16} or {16x8, 8x16} is made.

*3) 3rd level:* Inside the HIGH COMPLEXITY bin, a decision between {8x8, 8x4, 4x8} or {4x4, 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. 7 where the white circles represent the set of MB partition where the reference standard can choose into.

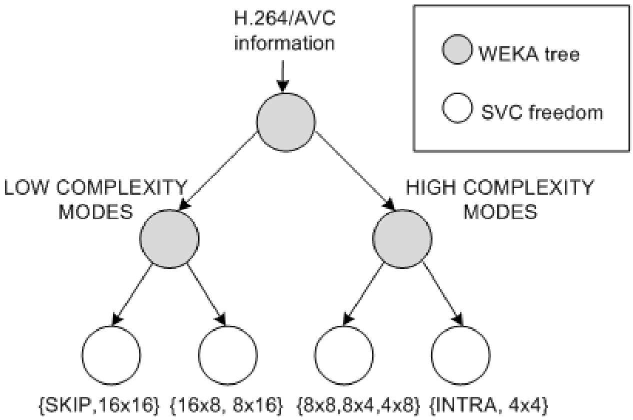


Fig. 7. Decision tree

The ML process gave us a decision tree that classified correctly in about 87% of cases in the 1st level, 80% in the 2nd level and 93% in the 3rd level as is shown in Table 1:

Table 1. % of correct choice of MB group

| **Classification of MB groups**<br>**Mode decision equal to the standard (%)** | | | |
|---|---|---|---|
| **Sequence** | **1st level** | **2nd level** | **3rd level** |
| Hall | 96.83 | 97.27 | 93.25 |
| City | 92.34 | 82.35 | 88.60 |
| Foreman | 87.84 | 79.46 | 93.00 |
| Soccer | 88.25 | 86.50 | 88.88 |
| Harbour | 80.23 | 66.86 | 94.55 |
| Mobile | 79.14 | 67.00 | 99.24 |
| ***Average*** | *87.44* | *79.91* | *92.92* |

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 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.

## 4. 2 Reducing the Search Area

As said previously, the idea of motion estimation task consists of eliminating the temporal redundancy in a way to determine the movement of the scene. For this purpose, in H.264/AVC MVs between every MB or sub-MB 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 main challenge to overcome in this transcoding architecture is the mismatching between GOP sizes, GOP patterns and prediction structures. While the starting encoded bitstream in H.264/AVC is formed by IPPP GOP patterns without temporal scalability, the final SVC bitstream needs conforming 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 SVC ones as shown in Fig. 3 so the number of MVs associated to an H.264/AVC MB can be different from the number of MVs associated to the corresponding SVC MB as illustrated in solid line in Fig. 8.

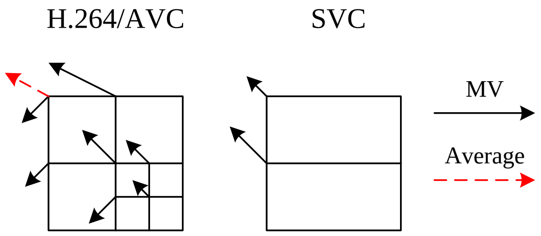


Fig. 8. Example MB in H.264/AVC with its MVs and the matching MB in SVC with its corresponding MVs

As Fig. 9 shows, there is not always a one-to-one mapping between previously calculated H.264/AVC MVs and the incoming SVC MVs. The present approach tries to tackle with this problem.

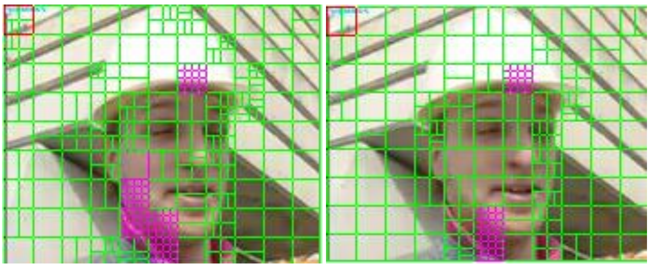


Fig. 9. MB partitions generated by H.264/AVC (left) and SVC (right) for the 2nd frame in the Foreman QCIF sequence

### *4.2.1 First stage: Initial Reduced Search Area*

The new reduced search area 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. 10.

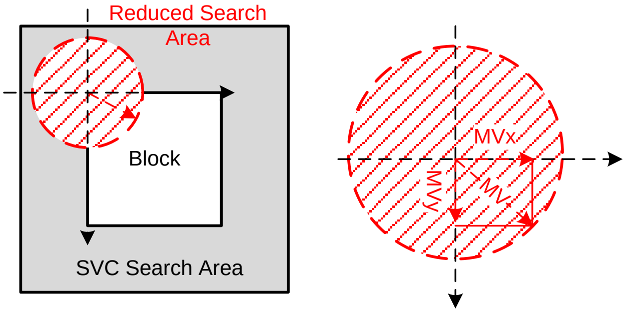


Fig. 10. Proposed reduced search area

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 (in dash line in Fig. 8) and the temporal layer which the frame is in. The average of the incoming MVs of a determined MB is used to overcome the situation explained previously where the number of MVs associated to a MB are different. The dependency of the layer will be explained in Section 4.2.2.

### 4.2.2 Second Stage: Adjusting the reduced search area

As it mentioned previously, MVs generated in H.264/AVC are re-used to generate a new small area defined by a circumference with the incoming MV for this MB as its radius.

Something 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. 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 it could seen in Fig. 2, with hierarchical pictures 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. This process is illustrated in Fig. 11.

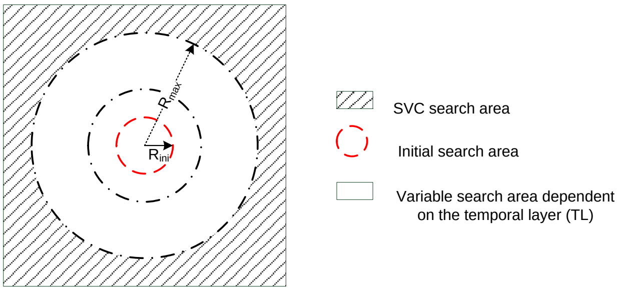


Fig. 11. Variation of initial search area depending on temporal layer

Here, coef depends on the number of the temporal layer (n) where the frame is in as defined in (1).

$$coef\,(n) = \frac{GOP_{length}}{2^n} \qquad (1)$$

## 5. Performance Evaluation

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 [14], version 16.2, with an IPPP pattern with 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, version 9.19.3 [15] with temporal scalability, Baseline Profile, different values of QP (28, 32, 36, 40) and GOP sizes of 4, 8 and 16.

For the results of our proposal, encoded bitstreams in H.264/AVC are transcoded using the technique described in Section 4. This technique was applied to the two enhancement temporal layers with highest identifier because, as it was shown in Fig. 12, those temporal layers is where most encoding time is spent. In these results, the training sequence Football has been excluded because it is not appropriate to test the results of a decision tree using the sequence that was used to generate it.

From Table 2 to Table 4 the results for ΔPSNR, ΔBitrate and Time Saving (TS) are shown when our technique is applied compared to the reference transcoder. ΔPSNR and ΔBitrate are calculated according to the Bjøntegaard-Delta metric [22].

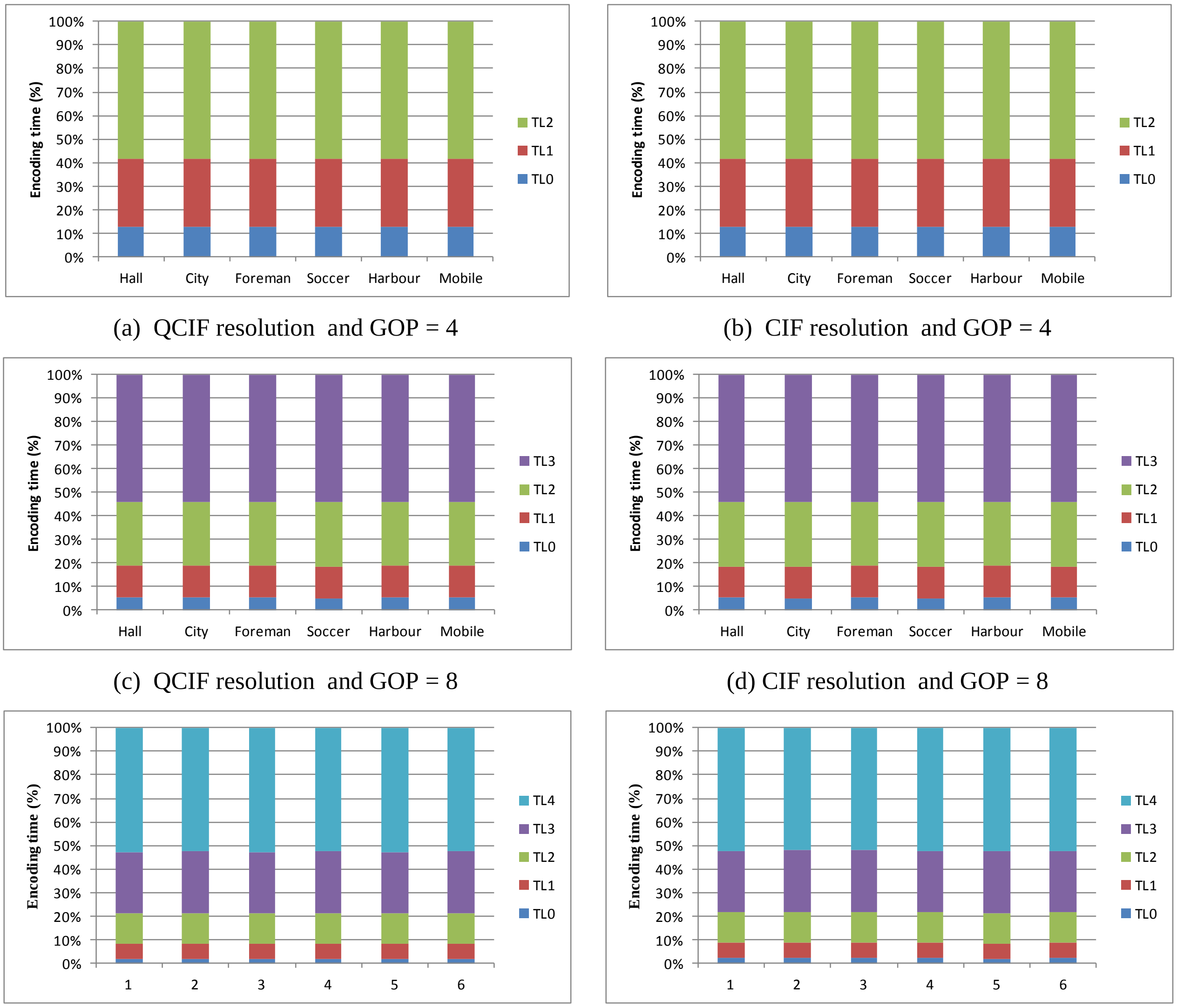
Time Savings are calculated for the full sequence (Full Seq.) and for the temporal layers where the technique is applied to (Partial). To evaluate it, (2) is calculated where $T_{ref}$ denotes the coding time used by the SVC reference software encoder and $T_{pro}$ is the time spent by the proposed algorithm.

$$Time\,Saving\,(\%) = \frac{(T_{ref} - T_{prop})}{T_{ref}} \cdot 100 \qquad (2)$$

ΔBitrate represents bitrate increase, ΔPSNR the difference in quality and a negative value means reduction and, finally, Time Saving represents complexity reduction for transcoding the bitstream.

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% of reduction of computational complexity in the full sequence and 98% in the specific layers is achieved. Moreover, our proposal is able to approach the RD-optimal transcoded (re-encoded) reference without any significant loss.

Fig. 13 shows the difference between the MB partitioning made by the reference transcoder and the proposed algorithm, with GOP = 4, CIF resolution and a QP value of 28 in sequences City, Soccer and Foreman. Those encoding processes were run under the same conditions.



(a) QCIF resolution and GOP = 4

(b) CIF resolution and GOP = 4

(c) QCIF resolution and GOP = 8

(d) CIF resolution and GOP = 8

(e) QCIF resolution and GOP = 16

(f) CIF resolution and GOP = 16

Fig. 12. Encoding time (%) in Baseline Profile for each temporal layer with different resolutions and GOP sizes

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** | **ΔBitrate** | **Time Saving (%)** | | **ΔPSNR** | **ΔBitrate** | **Time Saving (%)** | |
| | **(dB)** | **(%)** | ***Full Seq.*** | ***Partial*** | **(dB)** | **(%)** | ***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 | 99.27 |
| Foreman | 0.259 | 2.16 | 83.25 | 97.01 | -0.108 | 2.92 | 84.64 | 97.70 |
| Soccer | 0.037 | 2.51 | 81.77 | 94.52 | 0.022 | 2.30 | 82.58 | 95.60 |
| Harbour | 0.112 | -0.82 | 85.38 | 98.44 | 0.181 | -1.43 | 87.30 | 98.87 |
| Mobile | 0.151 | -0.17 | 84.33 | 98.19 | 0.246 | -2.30 | 85.24 | 98.24 |
| ***Average*** | ***0.141*** | ***0.92*** | ***84.46*** | ***97.74*** | ***0.146*** | ***0.26*** | ***85.61*** | ***98.13*** |

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

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**

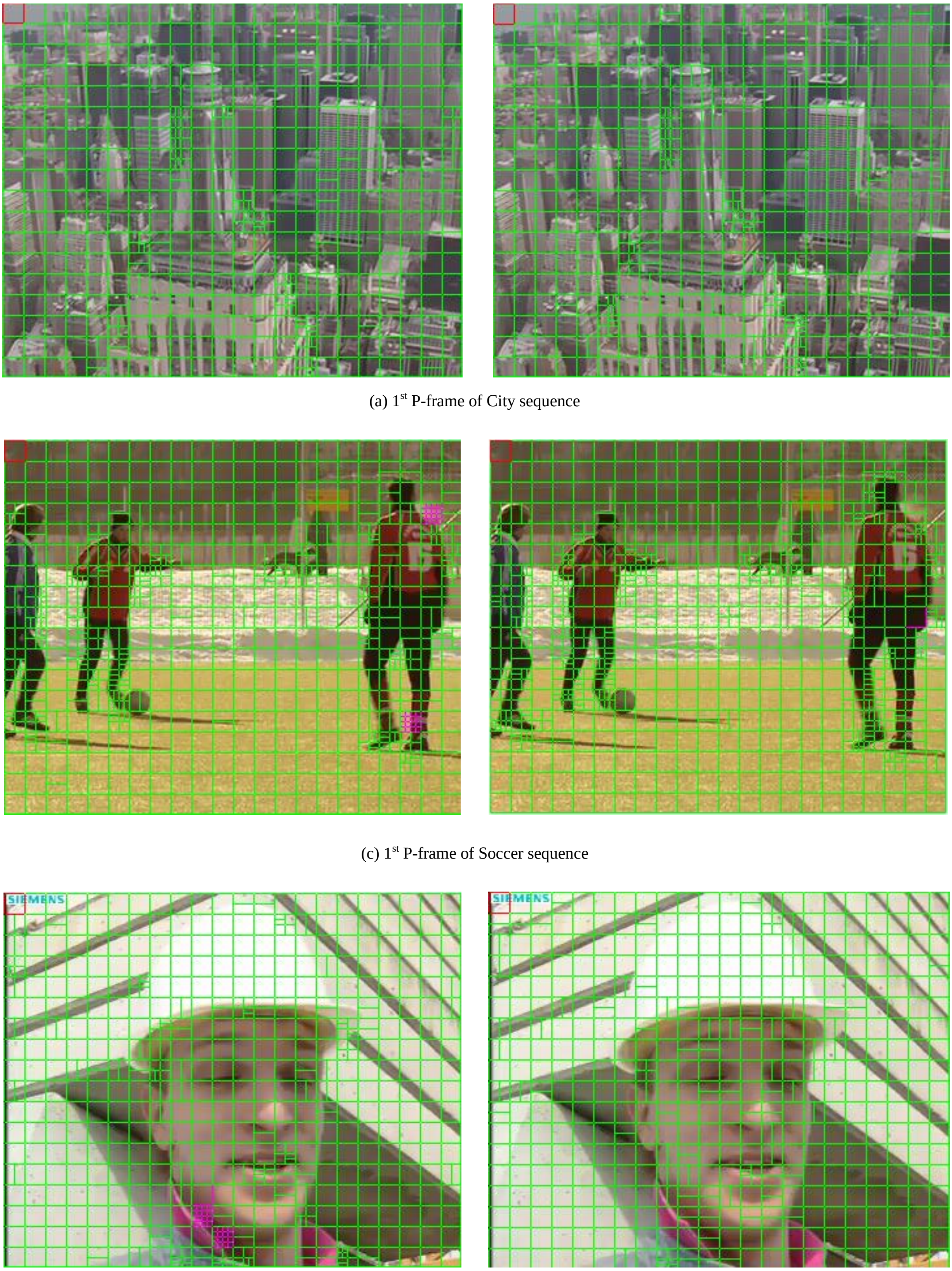
| | ***QCIF (15 Hz)*** | | | | ***CIF (30 Hz)*** | | | |
|---|---|---|---|---|---|---|---|---|
| **Sequence** | **ΔPSNR (dB)** | **ΔBitrate (%)** | **Time Saving (%)** | | **ΔPSNR (dB)** | **ΔBitrate (%)** | **Time Saving (%)** | |
| | | | ***Full Seq.*** | ***Partial*** | | | ***Full Seq.*** | ***Partial*** |
| Hall | 0.159 | 0.35 | 80.00 | 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 | 0.005 | 3.48 | 78.78 | 97.73 |
| Soccer | 0.066 | 2.96 | 75.45 | 94.49 | 0.000 | 2.55 | 76.96 | 95.67 |
| Harbour | 0.052 | 0.02 | 78.63 | 98.40 | 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*** | ***1.00*** | ***79.03*** | ***98.00*** |

Table 4. RD performance and time savings of the approach for GOP = 16 and different resolutions

**RD performance and time savings of H.264/AVC-to-SVC transcoder**

**GOP = 16**

| | ***QCIF (15 Hz)*** | | | | ***CIF (30 Hz)*** | | | |
|---|---|---|---|---|---|---|---|---|
| **Sequence** | **ΔPSNR (dB)** | **ΔBitrate (%)** | **Time Saving (%)** | | **ΔPSNR (dB)** | **ΔBitrate (%)** | **Time Saving (%)** | |
| | | | ***Full Seq.*** | ***Partial*** | | | ***Full Seq.*** | ***Partial*** |
| Hall | 0.327 | 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 | 1.90 | 76.22 | 99.10 |
| 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 | -2.60 | 75.68 | 98.46 |
| Mobile | 0.030 | 0.91 | 76.41 | 97.07 | 0.232 | -0.39 | 75.93 | 98.45 |
| ***Average*** | ***0.113*** | ***1.97*** | ***76.12*** | ***96.52*** | ***-0.060*** | ***1.50*** | ***75.42*** | ***98.06*** |



(a) 1st P-frame of City sequence

(c) 1st P-frame of Soccer sequence

(c) 1st P-frame of Foreman sequence

Fig. 13. MB partitioning for the proposed H.264-to-SVC transcoder  (left) compared to the reference one (right).

## 6. Conclusions

In this paper, a proposal for adapting 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 inter prediction process is reduced, and therefore, the complexity of the adaptation. The experimental results show that it is capable to reduce 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, 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] Advanced Television System Committee: ATSC-Mobile DTV Standard, A/153 ATSC Mobile Digital Television System. October 2009.

[2] 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.

[3] William W. Cohen: Fast Effective Rule Induction. In: 20th International Conference on Machine Learning, pp. 115-123, 1995.

[4] 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 Transaction on Multimedia vol. 11 n.7, pp.1209--1224, 2009.

[5] C. Develder, P. Lambert, W. Van Lancker et al., “Delivering scalable video with QoS to the home” Telecommunication Systems, vol. 49 n.1, pp. 129-148, 2012.

[6] 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.

[7] European Broadcasting Union: ETSI TR 102 377 V1.4.1: Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines. June 2009

[8] R. Garrido-Cantos, J. De Cock, J. L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido and R. Van de Walle, “Video Transcoding for Mobile Digital Television,”. Telecommunication Systems. Published online: DOI: 10.1007/s11235-011-9594-1.

[9] R. Garrido-Cantos, J. De Cock, J. L. Martínez, S. Van Leuven, and P. Cuenca, “Motion-Based Temporal Transcoding from H.264/AVC-to-SVC in Baseline Profile”, IEEE Transactions on Consumer Electronics, vol. 57, n. 1, 2011.

[10] R. Garrido-Cantos, J. De Cock, J. L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido and R. Van de Walle, “Low Complexity Adaptation for Mobile Video Environments using Data Mining”, 4th IFIP Wireless and Mobile Networking Conference (WMNC 2011), 2011.

[11] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, volume 11, n. 1, pp: 21-35.

[12] 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.

[13] Lian, S., “Secure service convergence based on scalable media coding”, Telecommunication Systems, vol. 45 n. 1, 2010.

[14] Joint Model JM Reference Software. http://iphome.hhi.de/suehring/tml/download/

[15] Joint Scalable Video Model (JSVM) Reference Software. http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm.

[16] Monteiro, J.M, Calafate, C.T, and Nunes, M.S, “Robust multipoint and multi-layered transmission of H.264/SVC with Raptor codes”, Telecommunication Systems, vol. 49, n.1, pp: 113-128, 2012.

[17] 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, Shangai, 2009.

[18] 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.

[19] H. Schwarz, D. Marpe and T. Wiegand, “Analysis of Hierarchical B pictures and MCTF,” IEEE Int. Conf. ICME and Expo, Toronto, 2006.

[20] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, n. 9, pp. 1103-1120, September 2007.

[21] P. Soohong and J. Seong-Ho, "Mobile IPTV: Approaches, Challenges, Standards and QoS Support", IEEE Internet Computing, vol. 13, n. 3, pp. 23--31, June 2009.

[22] 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

[23] S. Van Leuven, J. De Cock, G. Van Wallendael, R. Van de Walle, R. Garrido-Cantos, J.L. Martinez, and P. Cuenca, “A Low-complexity Closed-loop H.264/AVC to Quality-Scalable SVC Transcoder”, 17th International Conference on Digital Signal Processing , accepted for publication.

[24] S. Van Leuven, J. De Cock, G. Van Wallendael, R. Van de Walle, R. Garrido-Cantos, J.L. Martinez, and P. Cuenca, “Combining Open- and Closed-loop Architectures for H.264/AVC-to-SVC Transcoding”, 18th IEEE International Conference on Image Processing , in press.

[25] G. Van Wallendael, S. Van Leuven, R. Garrido-Cantos, J. De Cock, J.L. Martinez, P. Lambert, P. Cuenca and R. Van de Walle, “Fast H.264/AVC-to-SVC transcoding in a mobile television environment”, Mobile Multimedia Communications Conference, 6th International ICST, Proceedings, Lisbon, 2010.

[26] A. Vetro, C. Christopoulos and H. Sun, “Video Transcoding Architectures and Techniques: an Overview,” IEEE Signal Processing Magazine vol. 20 n.2, pp18--29, 2003.

[27] S. Wenger, “Temporal scalability using P-pictures for low-latency applications,” in IEEE Second Workshop on Multimedia Signal Processing, Redondo Beach, CA, USA, Dec. 1998, pp. 559–564.