Universidad de Castilla-La Mancha

Departamento de Sistemas Informáticos

TEMPORAL VIDEO TRANSCODING FROM H.264/AVC-TO-SVC

TESIS DOCTORAL

Presentada por: María del Rosario Garrido Cantos

Dirigida por: Dr. José Luis Martínez Martínez
Dr. Pedro Cuenca Castillo
Dr. Antonio Garrido del Solo

Albacete, abril del 2012

Temporal Video Transcoding from H.264-to-SVC

A dissertation for the degree of Doctor of Philosophy in Computer Science to be presented with due permission of the Computing Systems Department from University of Castilla-La Mancha, for public examination and debate

---

Author: *María del Rosario Garrido Cantos*

Advisors: *Dr. José Luis Martínez Martínez*
Dr. Pedro A. Cuenca Castillo
Dr. Antonio Garrido del Solo

Albacete, April 2012

AGRADECIMIENTOS

Quiero darles las gracias a mis directores de tesis, Pedro, Antonio y Pepelu que me han ayudado y apoyado en todo momento. Sin ellos esta tesis no se hubiera hecho realidad.

A todos mis compañeros del RAAP, con los que he compartido estos casi tres años de buenos y algunos no tan buenos momentos y los desayunos juntos de cada mañana. I would also like to thank the people from Multimedia Lab in Ghent, especially to Jan and Sebastiaan for helping me during the months I spent there and their remarks and ideas for the papers.

A mi familia que siempre ha estado ahí y que, al igual que yo, nunca se hubieran podido imaginar que acabaría haciendo una tesis doctoral. Por último, y no menos importante, agradecerle a Javi, por recorrer conmigo este camino, apoyándome y animándome tanto en los buenos, como en los malos momentos.

RESUMEN

En los últimos años la demanda de contenidos multimedia y de terminales para visualizarlos ha crecido espectacularmente. Por un lado, esos contenidos están codificados para reducir la capacidad de almacenamiento necesaria y el consumo de ancho de banda al transmitirlos. Por otro lado, las redes sobre las que se transmiten estos contenidos son heterogéneas, al igual que los terminales de los usuarios que presentan diferentes características en cuanto a ancho de banda, resolución de la pantalla, capacidad de procesamiento, etc.

Actualmente, la mayoría de contenidos multimedia codificados lo están bajo la norma H.264/AVC, con unas determinadas características como frames por segundo, resolución o calidad y, por tanto, existe una peor flexibilidad a la hora de adaptarse a diferentes anchos de banda o dispositivos. Sin embargo, esta adaptación del video sí que es posible si el contenido está codificado usando esquemas de codificación escalable. Uno de esos esquemas es SVC, una extensión a H.264/AVC

SVC proporciona escalabilidad temporal, espacial, de calidad o una combinación de las tres. Eso es posible gracias a su estructura de organización en capas (una capa base y una o varias de mejora). La capa base representa la menor tasa de transmisión, resolución y calidad y cada capas de mejora aumenta la tasa de frames por segundo, la resolución y la calidad. Eliminando capas de mejora del flujo de datos codificado se consigue adaptar el ancho de banda del canal o a las características del dispositivo a la vez que se permite que el vídeo sea totalmente decodificable.

Para que los contenidos existentes ya codificados según H.264/AVC sin ningún tipo de escalabilidad puedan beneficiarse de esta funcionalidad añadida es necesario un transcodificador eficiente de H.264/AVC a SVC que realice dicha transcodificación más rápido que si se decodificara y volviera a codificar de nuevo cada secuencia. Esta aceleración es posible reutilizando información de la etapa de decodificación de H.264/AVC para acelerar partes de la codificación en SVC. En esta tesis se proponen varias técnicas para acelerar dicha codificación en el marco del desarrollo de un

transcodificador de H.264/AVC a SVC con escalabilidad temporal. Para ello se hacen varias propuestas:

- *Reducción de complejidad del proceso de Estimación de Movimiento*: Antes de abordar este problema, se ha realizado un análisis previo del tiempo que necesita el codificador de SVC para codificar cada capa temporal para centrar la aplicación de las técnicas propuestas en aquellas capas en las que se consumiera más tiempo. Una vez realizada esa evaluación, se propone una técnica que emplea los vectores de movimiento procedentes del decodificador de H.264/AVC para reducir dinámicamente el área de búsqueda en SVC. Esta propuesta se basa en que dichos vectores representan, aproximadamente, la cantidad de movimiento existente en una escena. Tras aplicar dicha técnica para acelerar la parte del codificador del transcodificador propuesto, los resultados obtenidos muestran que se consigue una reducción considerable de tiempo, con un pequeño de aumento de bitrate y una pequeña pérdida de calidad.

- *Reducción de complejidad del proceso de Decisión de Modo:* Otro de los procesos que consume gran parte del tiempo en la transcodificación de la secuencia es la decisión de modo. Para reducir la complejidad de dicho proceso se propone un mecanismo para acelerarlo basado en el uso de técnicas de Data Mining. Basándose en la existencia de una correlación entre las particiones seleccionadas por SVC y cierta información extraída de H.264/AVC (residuo, vectores de movimiento, modos...), se emplean técnicas de Machine Learning para crear árboles de decisión que permitan, en función de la información extraída al decodificar una secuencia en H.264/AVC, seleccionar la mejor partición en SVC. Esto convierte un proceso muy complejo como es la decisión de modo en un árbol de decisión. De esta forma, se reduce significativamente la complejidad del codificador de SVC, tal y como demuestran los resultados obtenidos.

- *Reducción de la Predicción Inter:* Las dos propuestas mencionadas anteriormente han sido aplicadas conjuntamente, tanto en Baseline como en Main Profile y diferentes tamaños de GOP obteniendo una reducción muy significativa de la complejidad de la predicción Inter llevada a cabo en el codificador de SVC.

Estas técnicas para reducir la complejidad permiten implementar un transcodificador eficiente de H.264/AVC a SVC con escalabilidad temporal. Como se ha demostrado en esta tesis, pueden ser aplicadas por separado de manera conjunta para la reducción de la predicción inter, siendo válidas para diferentes perfiles y tamaños de GOP. Además, mejoran las técnicas propuestas hasta la fecha en la literatura.

SUMMARY

In the last years the demand of multimedia contents and terminals for visualizing has grown spectacularly. Normally, those contents are encoded to reduce the storage capacity needed and the bandwidth consumption when they are transmitted. The networks over which these contents are transmitted are heterogeneous, as well as the users' terminals which have different characteristics in terms of bandwidth, screen size, processing power, etc.

Nowadays most of video contents are encoded in H.264/AVC with an specific frame rate, resolution or quality and, therefore, they cannot be adapted to different bandwidths or devices. However, this video adaptation is possible if the content is encoded using scalable encoding schemes. One of those schemes is SVC.

SVC provides temporal, spatial and quality scalability or a combination of them. This is possible because the SVC bitstream is organized in layers (one base layer and one or more enhancement layers). The base layer represents the lowest frame rate, resolution and quality and every enhancement layer increments frame rate, resolution and quality. By removing enhancement layers from the encoded bitstream an adaptation to the cannel bandwidth or characteristics of the device is achieved

For existing contents already encoded in H.264/AVC without any scalability can benefit from this scalability is necessary an efficient H.264/AVC-to-SVC transcoder to perform this transcoding faster than if the video is decoded and fully re-encoded again in SVC. This is possible by reusing the information collected from the H.264/AVC decoding stage for acceleration parts of the SVC encoding. In this thesis, several techniques for accelerating those parts in an H.264/AVC-to-SVC transcoder with temporal scalability are proposed:

- *Motion Estimation complexity reduction:* Before addressing this issue, a preliminary analysis of the time needed by the encoder to encode each temporary layer was performed. The goal of this time analysis was found out in which

temporal layers most time was spent for focusing the application of the techniques presented in this thesis in those layers. Once made this evaluation, a technique that uses the ME vectors from the H.264/AVC decoding stage to dynamically reduce the search area in SVC is presented. This proposal is based on these vectors represent approximately the amount of motion present in a scene. After applying this technique to accelerate the encoder of the proposed transcoder, the results show that achieves a considerable reduction of time, with a slightly increase in bitrate and loss of quality.

- *Mode Decision complexity reduction:* Other process that consumes much of time in the encoding stage in SVC is mode decision process. To reduce the complexity of this process a mechanism for accelerating it based on data mining techniques is presented. Based on the existence of a correlation between the partitions selected by SVC and some information from H.264/AVC (residual, MVs, modes…), data mining techniques are used for developing decision trees that allow, based on the information extracted by the H.264/AVC decoding, selecting the best SVC partition. This technique becomes a very complex process in a simple decision tree and significantly reduces the complexity of the SVC encoder as shown by the results presented.

- *Interprediction complexity reduction:* The two proposals mentioned above have been applied together, in both Baseline and Main Profile and using different GOP sizes and resolutions, obtaining a significant complexity reduction of the interprediction process.

These presented techniques allow implementing an efficient H.264/AVC-to-SVC transcoder with temporal scalability. As demonstrated in this thesis, can be applied separately and together and are valid for different profiles and GOP size. Moreover, they improve the techniques proposed to date in the literature.

CONTENTS

FIGURE INDEX

TABLE INDEX

CHAPTER 1

INTRODUCTION

In this chapter, a brief introduction to the motivation and objectives of this thesis is done. Moreover, the thesis organization is shown.

1.1 Motivation

Mobile media services are being introduced on a daily basis into the market place in response to increasing user demand for ubiquitous media services and applications. Media contents are now being delivered over a wide variety of wireless/wired networks to mobile/fixed devices ranging from smartphones, tablets to powerful laptops or HDTV. Even some network technologies have been deployed specifically for delivering this content such as Mobile Digital TV networks: The newest one, ATSC-M/H [1] has been standardized in October 2009 and provides Mobile Digital TV service at United States of America, Canada, part of South America and part of Asia. Other network technology is DVB-H [2] that it has been standardized in November 2004 and adopted by European Commission in March 2008 as preferred technology to deliver Mobile Digital TV. This technology is used in Europe, part of Africa, Asia and Oceania. Both networks technologies are extensions of their terrestrial network technologies to deliver terrestrial digital TV. Another technology is MBMS [3] that uses the GSM / UMTS networks.

Unlike some years ago, we are confronted daily with video fragments and movies either on the Internet or TV. Several examples of devices receiving multimedia content are shown in Figure 1.1. The progress was made possible, to a large extent, by efficient image/video compression techniques that allow reducing the amount of data for being stored and transmitted and therefore dropping the resources required for that while keeping a quality image. MPEG-2 [4] video, which was standardized in the early nineties, and the MPEG-4 Visual [5] format (whose breakthrough was reinforced by the DivX [6] and XviD [7] implementations) have fostered the proliferation of video fragments and digitized movies. More recently, H.264/AVC [8] has been standardized. The H.264/AVC standard further reduces the video bitrate at a given quality when compared to previous specifications, and can be considered as the reference in video compression.

Figure 1.1. Examples of devices receiving multimedia content [9][10][11]

During encoding of media streams it is important to take into account the huge diversity of decoders and players. Multiple devices like PCs, laptops, smartphones, PDAs or HDTV are often used to play a single video file. Obviously, these devices have widely varying characteristics, which should be considered when sending media contents. Moreover, reliable reception of video contents by the mobile devices poses additional constraints because of the dynamic nature of the links and the limited resources of the mobile reception devices. In order to be able to deliver the media streams to the widest possible audience, a media communication system should be able of adapting on-the-fly the media streams to the transmission constraints and characteristics of the end-user devices to ensure high quality image continuously. Such adaptive media communications services are highly relevant for the development of efficient media consumer applications.

Additionally, the diversity of coding standards and formats used in production environments, distribution networks, and broadcast channels necessitate efficient media manipulation techniques. This diversity explains the necessity of media adaptation techniques. One way for easy stream adaptation is by using scalable coding schemes. Although provisions for scalable coding were already available for MPEG-2 and MPEG-4 Visual, they are rarely used in practice. Recently, MPEG and VCEG have standardized a new scalable extension of H.264/AVC that is denoted as SVC [12]. SVC allows creating scalable streams with minimal quality loss for the same bitrate when compared to single-layer H.264/AVC, providing different types of scalability such as temporal, spatial and quality or a combination of them in a flexible manner. This scalability is possible by creating a layered representation of the media stream during encoding process where the video is encoded as one base layer and one or more enhancement layers. The base layer contains the lowest frame rate (temporal scalability), the lowest resolution (spatial scalability) and the lowest quality (SNR or quality scalability). The enhancement layers provide information for increasing frame rate, resolution or details and fidelity. To remove redundancy between layers inter-layer prediction mechanisms are applied. This scalable media coding is an important mechanism not only to provide several types of end-user devices with different versions of the same encoded media stream, but it also enables its transmission at various bitrates. The bitstream is adaptable to the channel bandwidth or the terminal capabilities by truncating the undesired enhancement layers.

Despite these scalability tools, most of the video streams today are still created in a single-layer format (most of them in H.264/AVC) so these existing video streams cannot benefit from the scalability tools in SVC. Due to the fact that the migration from H.264/AVC to SVC is not trivial given the relatively high computational complexity of the SVC encoding process, it is likely that the dominance of single layer encoders will continue to exist in the near future. However, transcoding techniques exist that can make this process more efficient. Transcoding can be regarded as a process for efficient 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 the quality loss due to the adaptation.

Based on this challenge, the goal of this thesis is to develop an efficient H.264/AVC-to-SVC transcoder able to transform an H.264/AVC bitstream into an SVC bitstream faster than a cascade transcoder. Its efficiency is obtained by reusing as much information as possible from the original bitstream, such as mode decisions and motion information. The contributions focus on SVC with temporal scalability that allows varying the frame rate of the bitstream.

One possible application of this proposed transcoder could be to introduce it in the broadcaster side in a Mobile Digital TV network to transform already encoded content in H.264/AVC in SVC content (see Figure 1.2). In this way, some of these networks technologies (ATSC-M/H and DVB-H systems) have established recently a set of video coding specifications where H.264/AVC and SVC are chosen to transmit video in these networks and they also have defined the RTP packetization for video elementary streams [13][14].

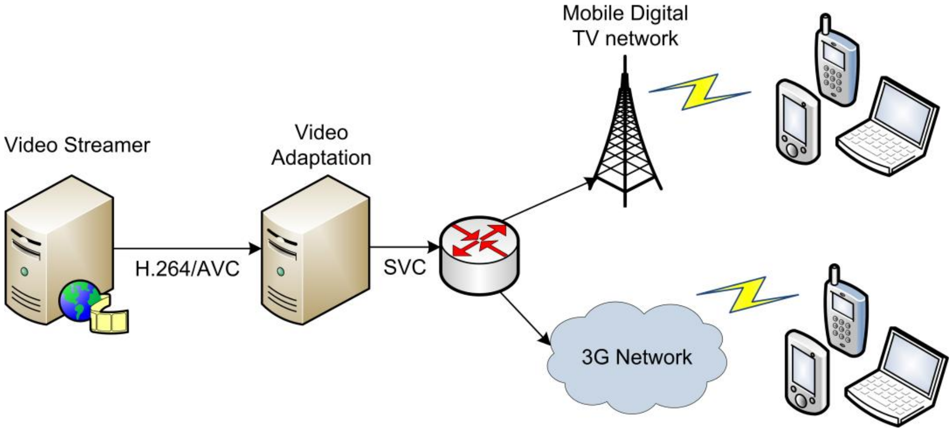


Figure 1.2. Example of an SVC transcoder for mobile environments

1.2 Objectives

As was mentioned previously, the main objective of this Thesis is to design an efficient H.264/AVC-to-SVC transcoder to convert bitstreams encoded using H.264/AVC without scalability into bitstreams encoded in SVC with temporal scalability. This transcoder needs to be able to work faster than the reference transcoder (a cascade transcoder that decodes and encodes completely the bitstream) while maintaining coding efficiency. Some partial objectives defined to achieve the goal are details below:

- Making a study of H.264/AVC and SVC. This study will allow extracting the similarities and differences of both video standards.
- Studying the existing techniques of video transcoding, focusing on the state-of-the-art of the proposals related to H.264/AVC-to-SVC transcoding which is the goal of this thesis. This study set the basis and provides an overview of the ideas already developed.
- Identifying the parts which can be accelerated in an H.264/AVC-to-SVC transcoder and the time consuming for encoding every temporal layer.
    - As in H.264/AVC, two of the most time consuming tasks in the SVC encoder are ME and mode decision.
    - Moreover, an analysis of the time spent for the SVC encoder for encoding every temporal layer will be done. The results of this study will allow focusing in the temporal layers where the encoder spends more time applying the proposals to these temporal layers.
- Proposing some algorithms to accelerate the tasks identified previously, maintaining coding efficiency. These algorithms will be applied to the temporal layers which need more time for being encoding and will be developed reusing

information from the H.264/AVC decoding stage for reducing the time needed by the encoding stage. These techniques will be used for accelerating ME and mode decision tasks.

  - For accelerating ME, a way for reducing the search area to be checked can be proposed reusing motion information available after decoding.
  - For reducing the time spent in the encoder in mode decision task, a faster algorithm based in decision trees can be implemented taking into account the correlation between the information collected from H.264/AVC decoding stage and the SVC decision modes.

- Developing a H.264/AVC-to-SVC transcoder using together the algorithms proposed in order to provides temporal scalability in different video with different resolutions, encoding using different profiles and a large range of GOP sizes.
- Evaluating the algorithms proposed. Performing a wide range of tests for evaluating the results of the techniques presented to ensure that these proposals work faster than the reference transcoder while maintaining efficiency.
- Improving the techniques existing at this moment for transcoding from H.264/AVC-to-SVC. For that, a comparison between the results obtained during this thesis and the existing at this moment will be done.

1.3 Thesis Organization

This thesis is organized in seven chapters, which are introduced here:

- *Chapter 1*. The introduction chapter briefly describes the accomplished work. Motivations, objectives and organization of the document are also described in this chapter.
- *Chapter 2*. In this chapter a review of the video compression standards used in this thesis (H.264/AVC and SVC) was made.
- *Chapter 3*. In this chapter are summarized the typical transcoding architectures, focusing on the state-of-the-art of H.264/AVC-to-SVC transcoding since it constitute the main part in this thesis.
- *Chapter 4*. In this chapter, a technique for acceleration the ME task of the encoding stage in a H.264/AVC-to-SVC transcoder scenario is presented.
  This technique proposed collects some information from the decoding stage related and reused for reducing the search area and therefore for reducing the necessary encoding time while maintaining coding efficiency.
- *Chapter 5*. In a H.264/AVC-to-SVC transcoder scenario, a technique for accelerating the mode decision task of the encoding stage is presented. This algorithm uses Data Mining techniques to find relationships between the information collected from the decoding stage and the MB partitioning selected by the encoding stage for building a decision tree. Then, this decision tree is used

to reduce the number of MB types that the encoding stage has to check. Thus, the encoding time is reduced significantly while the coding efficiency is maintained.

- *Chapter 6*. It presents a joint proposal using the techniques presented in Chapter 4 and 5. An H.264/AVC-to-SVC transcoder with a reduction of a 75% of encoding time is achieved. As in the previous chapters, a complete performance evaluation with different GOP sizes and video profiles is presented. Moreover, a compilation of all the experimental results of the thesis and comparisons with other algorithms are also shown.
- *Chapter 7*. In this chapter are presented the conclusions, future work and publications derived from this thesis.

CHAPTER 2

H.264/AVC AND SVC

Since H.264/AVC and SVC are the standards involved in the development of this thesis, they will be described in this chapter.

2.1 Introduction

In the last years, the number of multimedia contents has grown exponentially. These contents need to be compressed for easy storage and transmission, making video compression techniques in essential. Since 1984, a wide range of digital codecs have been standardized. This standardization sets restrictions on bitstream, bitstream syntax and decoder operation as shown in Figure 2.1.

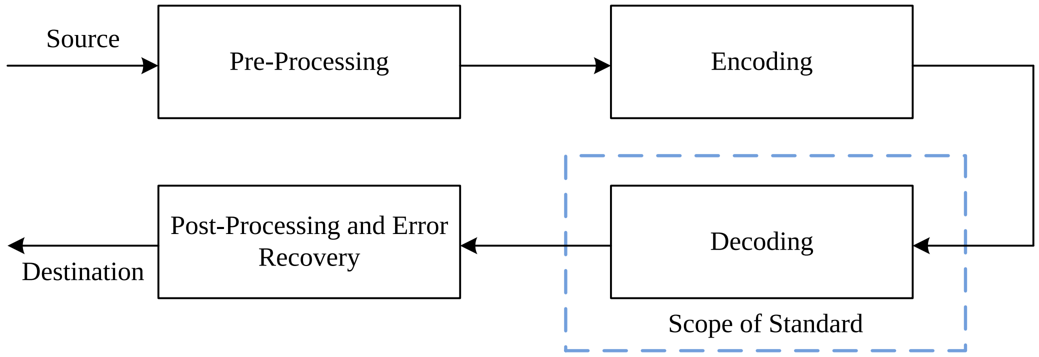


Figure 2.1. Scope of video coding standardization

At this moment there are two organizations that define video codec standards.

- *International Organization for Standardization*. In the ISO, there is a working group named MPEG that publishes the video and audio compression standards.

- *International Telecommunication Union*. The ITU-T produces standards for being use in communications. As in the ISO, there is a group specialized in standardize video compression standards. This working group is named the VCEG and was the responsible of publishing H.26x series of video coding standards.

A few years ago, a group of video coding experts from these groups was created to develop advanced video coding specifications and it is named as JVT. An illustration of the historical evolution of video coding standards is shown in Figure 2.2.

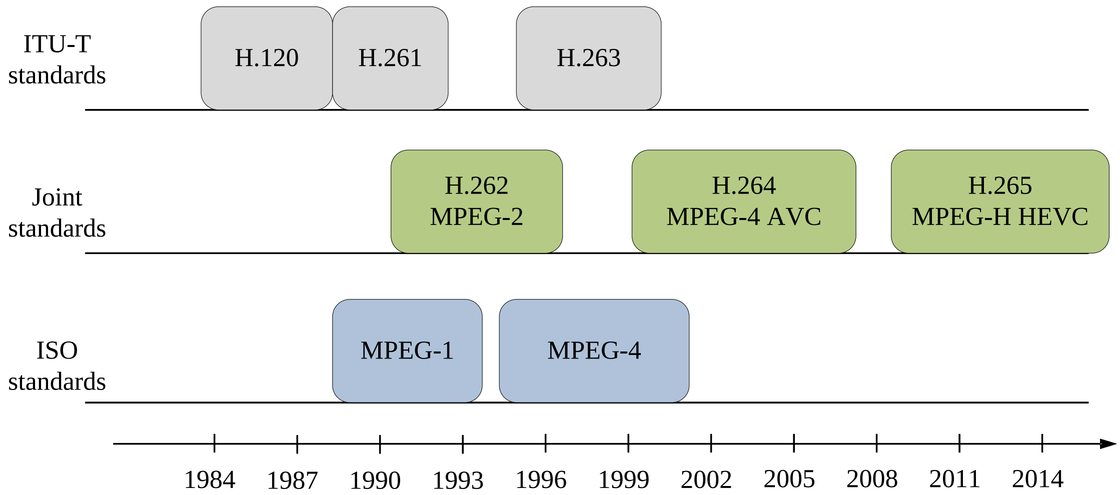


Figure 2.2. Evolution of video coding standards

The first digital video codec named H.120 [15] was standardized in 1984 by the ITU-T. The first version of the codec featured some very basic compression techniques. An updated version was produced in 1988.

H.261 [16] was standardized in 1990 by ITU-T and is widely considered to be the codec on which most modern video coding standards are based. It was developed to support videotelephony and videoconferencing over ISDN circuit-switched networks.

In early '90s, the first video coding standard developed by the ISO was standardized. It was the MPEG-1 [17] and was based in H.261 although a number of coding efficiency enhancement was added. This standard was developed for storing VHS quality video on media such as CD-ROMs.

Later, in middle '90s, the first joint standardization between ITU-T and ISO called H.262 [17] or MPEG-2 was developed by the JVT. This standard provided a solution for digital TV broadcasting via cable, satellite and terrestrial channels. Moreover, it was also employed to store multimedia contents in DVDs.

After that joint standard, ITU-T developed H.263 [18] and, at around the same time, ISO standardized MPEG-4 [5]. H.263 was designed as a successor to H.261 and was capable of providing better compression efficiency than H.261. Two further evolutions of this standard with improved compression efficiency where named as H.263+ and H.263++. Unlike the previous MPEG standards, MPEG-4 was not aimed at any particular application. This standard had the ability to describe a video scene as a number of objects. One issue with this video codec is that offers very little gain in compression efficiency compared to H.263+.

In 2003, JVT developed a new standard that improves the MPEG-4 and the H.263 standards, providing better compression of video sequences. The new standard is known as AVC and it was published jointly as Part 10 of MPEG-4 and ITU-T Recommendation H.264/AVC [8]. H.264/AVC provides the mechanisms for coding video which are optimized for compression efficiency and are aimed at satisfying the needs of practical multimedia communication applications. This new standard offers more choices of coding parameters and strategies than MPEG-4. It provides enhanced coding efficiency for a wide range of applications, including video telephony, videoconferencing, TV, storage (DVD and/or hard disk based, especially high-definition DVD), streaming video, digital video authoring, digital cinema and many others. Therefore, since at present, H.264/AVC is the best compression standard an also it is the most referenced one.

In 2007, an extension of H.264/AVC video compression standard name as SVC [12] was proposed by JVT. It supports spatial, temporal and quality scalability for video that allows on-the-fly adaption to certain application requirements such as display and processing capabilities of target devices and varying transmission conditions. These characteristics becomes SVC in a highly attractive solution to deal to the wide range of applications that used video coding nowadays which have variable transmission conditions, devices with heterogeneous display and computational capabilities (see Figure 2.3).

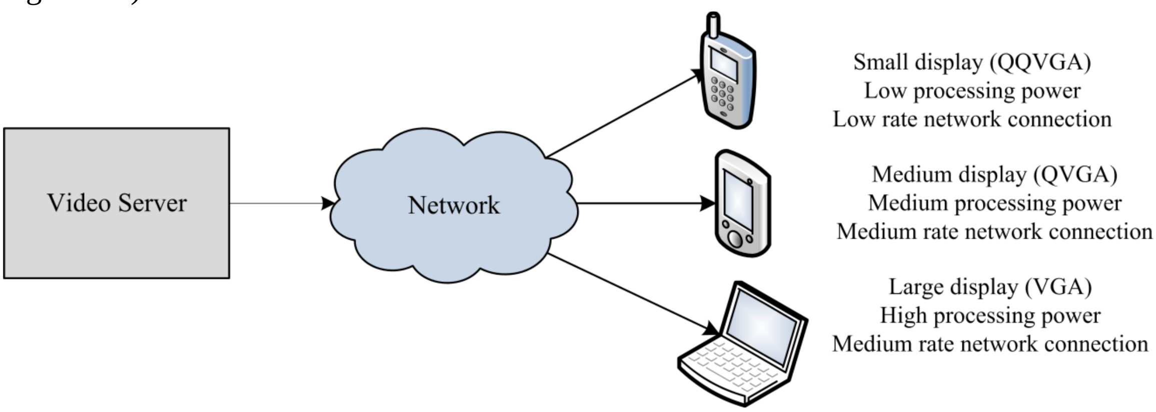


Figure 2.3. Possible scenario for SVC application

Currently, the standardization process of another JVT video codec is taking place. The new video coding standard will name H.265 [19]. In the following sections H.264/AVC and its SVC extension are described.

2.2 H.264/AVC

H.264/AVC has higher compression capability than the previous video coding standards. One of the main goals in the development of H.264/AVC was to manage the needs of many different video applications and delivery networks that could be used to carry the coded video data. The standard is divided into a VCL and a NAL. This structure is shown in Figure 2.4:

- VCL represents the coded source content.
- NAL formats the VCL representation of the video and provides header information appropriately for transportation by a variety of transport layers or storage media.

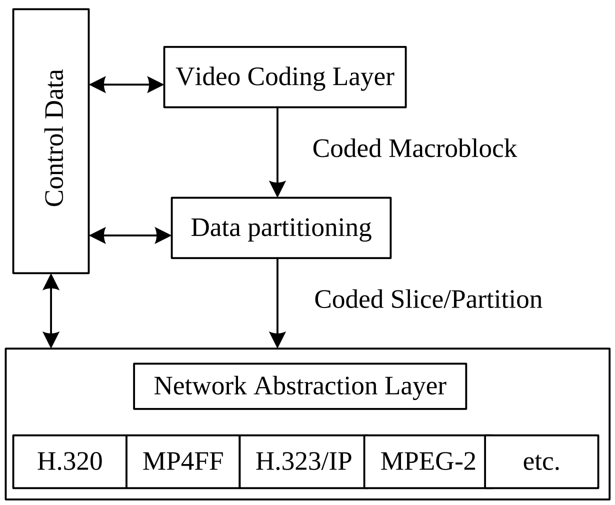


Figure 2.4. Structure of H.264/AVC video encoder

2.2.1 Network Abstraction Layer

As it was mentioned previously, NAL formats the VCL representation of the video and provides header information appropriately for transportation by a variety of transport layers or storage media such as RTP/IP, MP4, H.32X, MPEG-2, etc.

All data are contained in NAL units and each of which contains an integer number of bytes. A NAL unit starts with a one byte header which signals the type of the contained data. The remaining bytes represent payload of the contained data.

NAL units are classified into VCL NAL units which contain coded data and non VCL NAL units which contain associated additional information. A set of consecutive NAL units with specific properties is referred to as an access unit. The decoding of an access unit results in exactly one decoded picture. A set of consecutive access units with certain properties is referred to as a coded video sequence. A coded video sequence represents an independently part of NAL unit bitstream that can be decoded. It always starts with an IDR access unit, which signals that the IDR access unit and all following access unit can be decoded without decoding any previous pictures of the bitstream. For more information regarding NAL see [20][21][22].

2.2.2 Video Coding Layer

H.264/AVC employs a hybrid block-based video compression technique which is based on combining picture Interprediction to exploit temporal redundancy and transform-based coding of the prediction errors to exploit spatial redundancy. A H.264/AVC basic coding structure for a MB is shown in Figure 2.5.

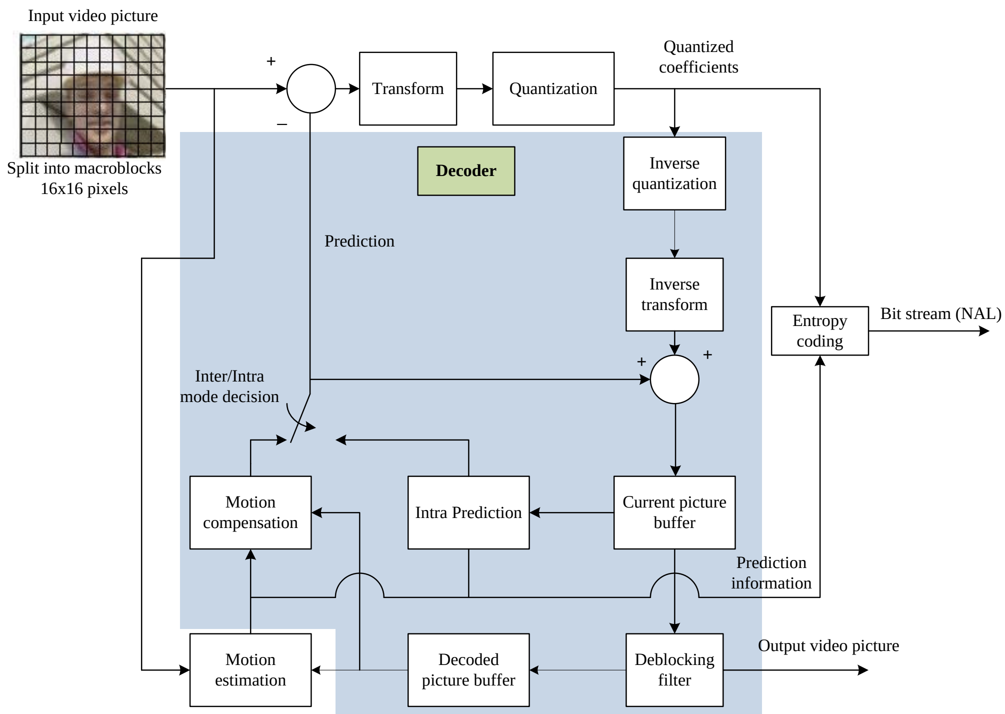


Figure 2.5. Basic coding structure for H.264/AVC for a MB

The input image is divided into macroblocks. Each MB is composed by three components: Y, Cr and Cb. Y is the luminance component (luma) and Cr and Cb represent the colour information (chroma). Human eye is more sensitive to luminance than chrominance, so Cr and Cb can be sub-sampled for obtaining different formats such as 4:2:0, 4:2:2, 4:4:4, etc.

These macroblocks are coded in Inter or Intra mode. In Inter mode a MB is predicted using ME and compensation. For this prediction, a displacement vector is estimated and transmitted for each block (motion data) that refers to the corresponding position of its image signal in an already transmitted reference image stored. In Intra mode, the image can be coded without reference to previously sent information since a MB is coded using the information of spatial neighbours.

The residual of the prediction, which is the difference between the original and the predicted MB, is transformed. Then, the transformed coefficients are scaled, quantized and entropy coded.

The encoder also decodes the picture to provide a reference for future predictions. The quantized transform coefficients are inverse scaled and inverse transformed in the same way as at the decoder resulting in the decoded prediction residual. This decoded prediction residual is added to the prediction. The result of that addition is introduced into a deblocking filter which provides the decoded video as its output. The main features of this coding scheme are explained in more detail in the following subsections.

Slices and Slices Groups

As it is explained previously, pictures in H.264/AVC are divided into macroblocks and each one is composed by luma pixels and chroma pixels. Each picture can be also divided into a number of independently decodable slices where a slice contains one or more macroblocks. The slices can be of different types and this type determinates which prediction modes are available for the macroblocks:

- I slice: All macroblocks are encoded in Intra mode.
- P slice: In addition to the coded types allowed in I slices, this type of slice can contain macroblocks coded using interprediction with one reference picture.
- B slice: In addition to the coded types allowed in P slices, this type of slice can contain macroblocks coded using interprediction with two reference pictures.
- SP and SI slices: They are specific slices that are used for an efficient switching between two different bitstreams. SP slices exploit motion compensated prediction and SI slices can exactly reconstruct SP ones. Slices of different types can be mixed in a single picture.

The partitioning of a picture into slices can be done in different ways depending on it is used the partition method of previous standards or the new partition method implemented in H.264/AVC. In previous standards, the shape of a slice was often constrained and macroblocks contained in the same slice were always consecutive in the order of a raster scan of the picture or of a rectangle within the picture as it is shown in Figure 2.6.

In H.264/AVC exists a more flexible way based on the concept of slice group as known as FMO. This method is supported only in some profiles of the standard (see subsection 2.2.3 for more information regarding H.264/AVC profiles).

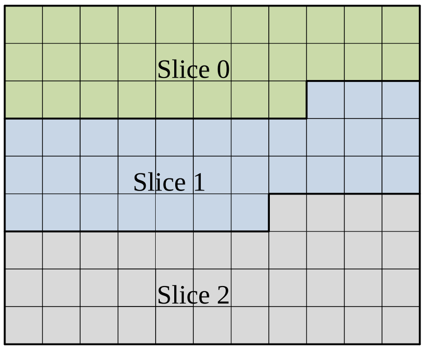


Figure 2.6. Picture divided into slices without using FMO

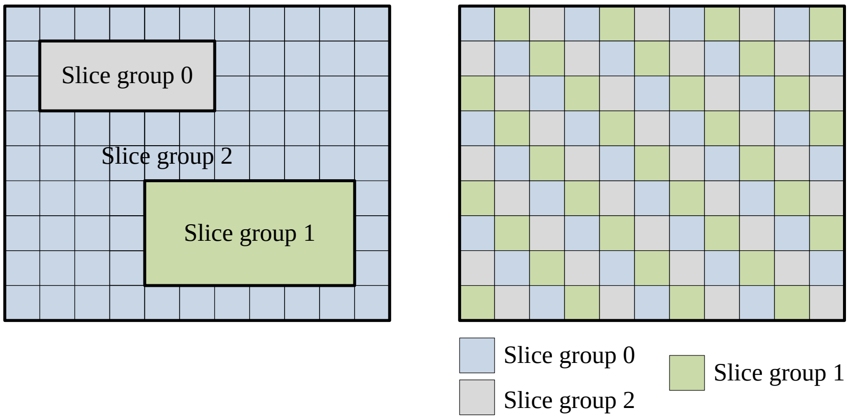


Figure 2.7. Examples of slice groups

Using the slice group concept, the allocation of macroblocks into slices can be made flexible through the specification of slice groups and MB allocation in picture parameter set. A MB allocation map is specified and specifies each MB of the picture which slice group belongs to. Each slice group can be divided into one or more slices, where each slice is composed by an integer number of macroblocks in raster-scan order within its slice group. Some examples are shown in Figure 2.7.

Intra Prediction

H.264/AVC incorporates into its coding process an intra prediction (defined within the pixel domain) whose main aim is to improve the compression efficiency of the intra-coded blocks. Intra prediction uses the information from neighbouring samples values of the current picture that have been already decoded and reconstructed for predicting individual sample values. This process is carried out at MB level.

For each MB and colour component (Y, Cr and Cb), one prediction mode and set of prediction directions have to be obtained. The H.264/AVC standard has three different intra prediction modes for the prediction of luminance component Y: Intra 4x4, Intra 8x8 and Intra 16x16. These modes correspond to the block size (4x4 pixels, 8x8 pixels and 16x16 pixels). Each intra prediction mode includes several directional predictions greatly improving the prediction in the presence of directional structures.

In the Intra 4x4 mode, each 4x4 luma block within a MB can use a different prediction mode. There are nine possible modes: DC and eight directional. For example, in the horizontal prediction mode, the prediction is formed by copying the samples immediately to the left of the block across the rows of the block. In Figure 2.8 it is shown several intra prediction modes for Intra 4x4 prediction.

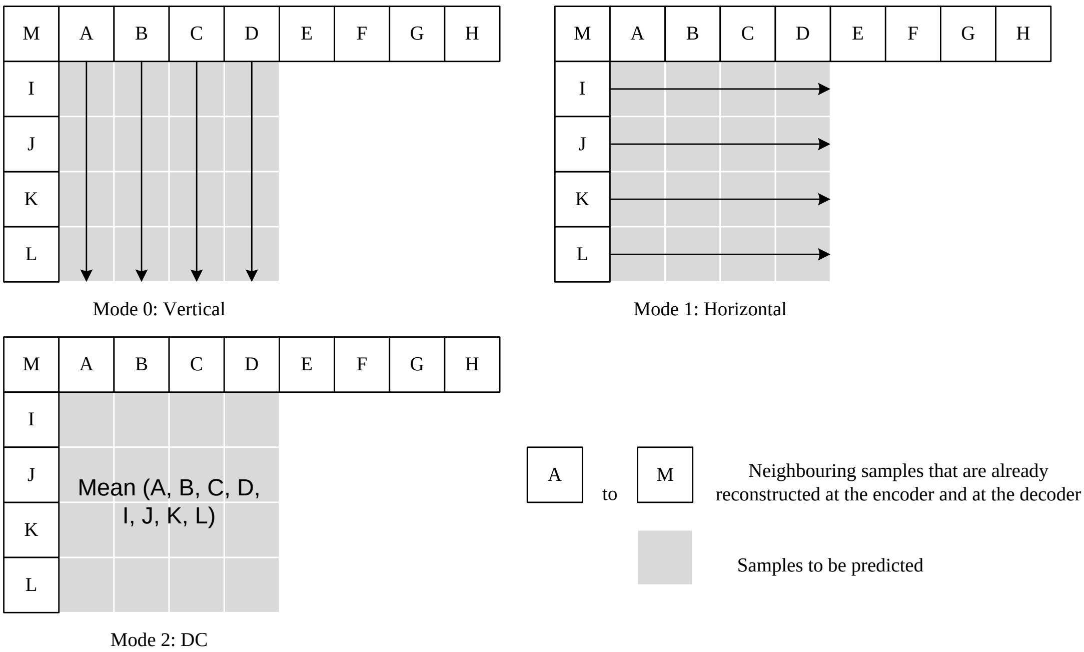


Figure 2.8. Three of the nine Intra 4x4 prediction modes

The nine possible intra prediction directions and the operations made with A to M samples to get the different prediction are summarized in Table 2.1. These predictions are calculated for each of the sixteen 4x4 blocks in a MB. The Intra 8x8 mode operates similarly, except that each 8x8 luma block has to be low-pass filtering before to use one of the nine prediction directions explained previously.

The Intra 16x16 mode also operates similarly, except that the entire luma component of the MB is predicted at once, based on the samples above and to the left of the MB. Also, in this mode there are only four modes available for prediction: DC, vertical, horizontal and planar. This new prediction direction is shown in Figure 2.9. For the chrominance components, the H.264/AVC standard only defines one mode for applied the intra prediction: the 8x8 Intra chroma prediction. Each 8x8 chroma block is predicted using the samples above and/or to the left. The chroma components (Cb and Cr) in a MB may have the same prediction mode. The prediction method is similar to Intra 16x16 luma prediction. The prediction mode for chroma components is selected independently of the prediction mode for luminance. Intra prediction and all other forms of predictions across slice boundaries are not used, in order to keep all slices independent of each other

Table 2.1. The nine prediction directions in Intra 4x4 luma prediction mode.

Direction	Description
0 (Vertical)	The upper samples A, B, C and D are extrapolated vertically.
1 (Horizontal)	The left samples I, J, K and L are extrapolated horizontally.
2 (DC)	All samples (a to p) are predicted by the mean of samples A to D and I to L.
3 (Diagonal Down-Left)	The samples are interpolated at a 45º angle between lower-left and upper-right.
4 (Diagonal Down-Right)	The samples are interpolated at 45º angle down and to the right.
5 (Vertical-Right)	Extrapolation at an angle of approximately 26.6º to the left of vertical.
6 (Horizontal-Down)	Extrapolation at an angle of approximately 26.6º below of horizontal.
7 (Vertical-Left)	Interpolation at an angle of approximately 26.6º to the right of vertical.
8 (Horizontal-Up)	Interpolation at an angle of approximately 26.6º above horizontal.

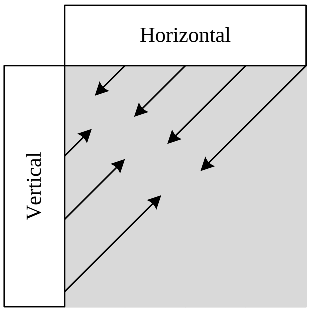


Figure 2.9. Intra 16x16 prediction planar mode

Motion Compensated Prediction

In this type of prediction (Interprediction), blocks of samples from previously reconstructed reference pictures are used to predict current blocks transmitting MVs.

For this purpose, each MB can be divided into smaller partitions (MB partitions). The luminance component (16x16 pixels) can be partitioned in four different modes as it is shown in Figure 2.10.

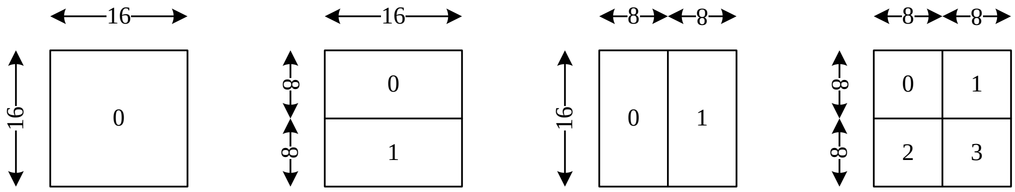


Figure 2.10. MB partition modes

If the 8x8 partition mode is chosen, each 8x8 MB partitions within the MB can be further partitioned in one of the four ways as shown in Figure 2.11. This method of partitioning macroblocks into motion compensated sub-blocks of varying size is known as tree structured MC.

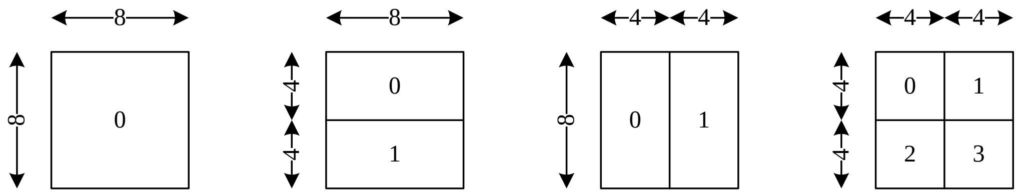


Figure 2.11. 8x8 block partition modes

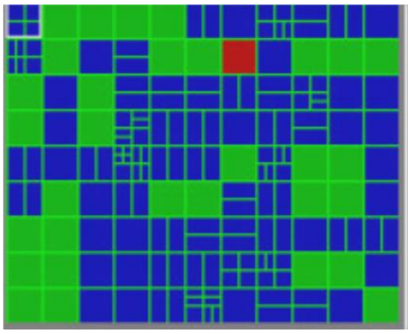
The resolution of each chrominance component in a MB (Cr and Cb) is half of luminance component. Each chrominance block is divided into the same way as luminance one, except that partition sizes have exactly half horizontal and vertical resolution (for example, 8x16 partition in luminance corresponds to 4x2 in chrominance). The horizontal and vertical components of each MV (one for each partition) are reduced by half when are applied to the chrominance blocks.

Figure 2.12 shows the second picture of Foreman, Flower and Paris sequences and their partition modes made by Interprediction in Baseline Profile with all parameters as default. In Figure 2.13 is shown the meaning of the different types of Inter macroblocks.

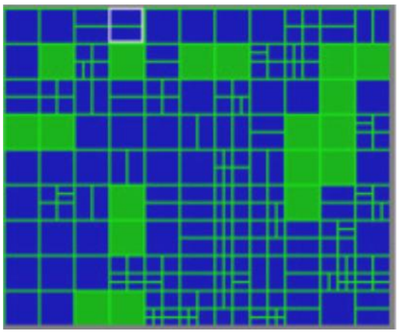
In order to evaluate the MVs, each partition in an inter-coded MB is predicted from an area of the same size in a reference picture. The offset between the two areas (the MV) has 1/4 pixel resolution for luminance component. If the video source sampling is 4:2:0, 1/8 pixel samples are required in the chrominance components (corresponding to 1/4 pixel samples in luminance). The luminance and chrominance sample at sub-pixel positions do not exist in the reference picture and so it is necessary to create them using interpolation from nearby image samples at integer locations. For example, in Figure 2.14 a 4x4 block in the current picture (Figure 2.14 a) is predicted from a region of the reference picture neighbouring to the current position. If the horizontal and vertical component of MVs are integer (Figure 2.14 b), the relevant pixel elements in reference block already exist (blue dots). If one or both vectors components are fractional values (Figure 2.14 c), the prediction samples (blue dots) are generated by interpolation between adjacent samples (white dots).

H.264/AVC supports MC with multiple reference frames, that is, more than one previously coded picture may be simultaneously used as prediction reference for the MC of the macroblocks in a picture (see Figure 2.15). To allow this, encoder and decoder store the reference frames in a memory with multiple frames. The MC prediction of each MB can be derived from one or more reference pictures of the buffer (not necessarily temporally consecutives) and it is possible to use weight prediction (in several profiles) where values of this region predicted can be multiplied by a weight.


(a) Foreman second picture


(b) Foreman second picture partition mode


(c) Flower second picture


(d) Flower second picture partition mode


(e) Paris second picture

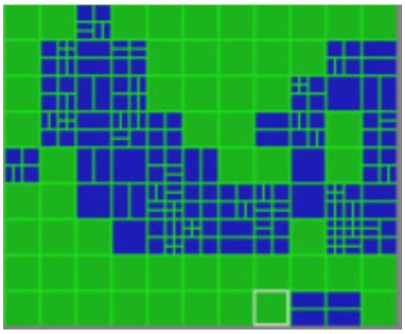

(f) Paris second picture partition mode

Figure 2.12. Interprediction in H.264/AVC

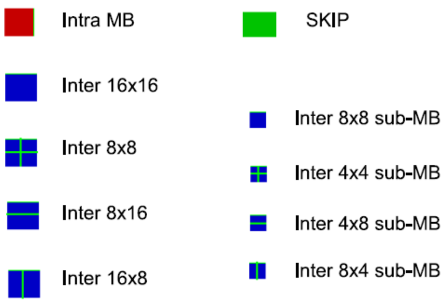
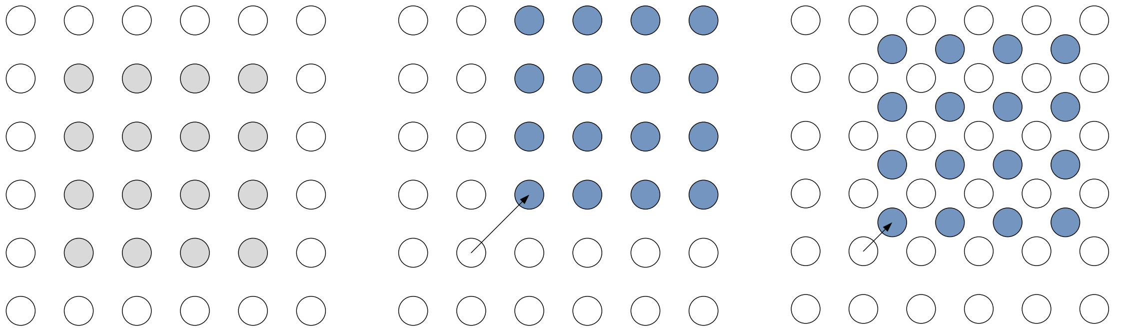


Figure 2.13. Different types of Inter MB in Figure 2.12

For example, it is possible that a reference picture used in a P slice is located temporally after the current picture or that both references for prediction a MB in a B slice be located before or after the current picture. This characteristic usually is quite useful when it is applied in situations like periodic motion, changes between different angles of camera focusing two different scenes and interpretation of movements.



(a) 4x4 block current frame (b) Reference vector (1,-1) (c) Reference vector (0.75,-0.75)

Figure 2.14. 4x4 example of integer and sub-sample prediction

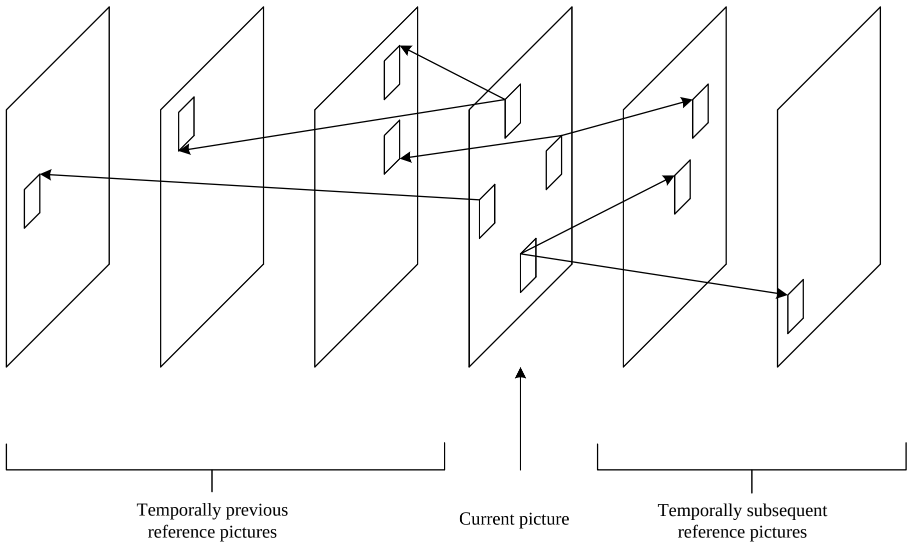


Figure 2.15. Multireference picture prediction in H.264/AVC

In-loop deblocking filter

Block based prediction and transform coefficients can lead to visible artefacts. To reduce their visibility, a filtering process is applied. This process is performed identically in both the encoder and decoder, in order to maintain an identical set of reference pictures. This filtering leads to improvements in quality as shown in Figure 2.16.

The basic idea of the filter is that a big difference between samples at the edges of two blocks should only be filtered if it can be attributed to quantization because if it is not like that, that different is coming from the image itself. The deblocking filter operates on 4x4 grid, which is the smallest basis for the block transform and interprediction. Both luminance a chrominance samples are filtered.

(a) Without deblocking filter (b) With deblocking filter

Figure 2.16. A decoded frame of Foreman [21] sequence without and with filtering

The filter is highly adaptive in three levels in order to remove as many artefacts as possible:

- *On slice level*: The filter strength may be adjusted to the characteristics of the video sequence.
- *On block edge level*: The filter strength is adjusted depending of the type of coding (Intra or Inter), the motion and coded residues.
- *On sample level*: The filter may be switched off depending on the type of quantization.

The filter is controlled through a parameter which defines the filter strength. Regarding order of filtering, it can be done on a MB basis, that is, immediately after a MB is decoded. First, the vertical edges are filtered, then horizontal ones. The bottom row and right column of a MB are filtered when decoding the corresponding adjacent macroblocks.

Transform, Quantization and Scaling

H.264/AVC uses transform coding to represent the prediction residual. The task of the transform is to reduce the spatial redundancy of the prediction residual. Different integer transforms are applied in H.264/AVC depending on the type of prediction residue to code. In the first version of the standard the available transform were:

- 4x4 HT for the luminance DC coefficients in macroblocks coded with the Intra 16x16 mode.
- 2x2 HT for the chrominance DC coefficients in any MB.
- 4x4 Integer Transform based on DCT for all the other blocks.

In a later amendment, transform based on blocks of 8x8 samples was introduced. The H.264/AVC transform is based on DCT, but with some fundamental differences:

- It is an integer transform which implies that no floating point operations are need.
- It can be implemented using only additions and shifts.
- The number of operation can be reduced by integration part of the operations involved in the transform into the quantizer.

As depicted in the H.264/AVC reference standard, the two dimensional DCT transform is implemented applying a one dimensional DCT transform twice, one to the horizontal dimension and another to the vertical one. In the first step, the horizontal correlation within the *nxn* samples block is exploited and in the second step, the one dimensional DCT transform is applied to exploit the vertical correlation. The transform matrices T defined in the standard are:

$$T_{8x8} = \begin{pmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & -8 \\ 6 & -12 & 3 & 10 & 10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{pmatrix} \qquad T_{4X4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}$$

Quantization removes irrelevant information from the pictures to obtain a rather substantial bitrate resolution. This process corresponds to the division of each coefficient by a quantization factor. In H.264/AVC, quantization is performed with the same quantization factor (Q_{step}) for all the transform coefficients in the MB. A total of 52 values for quantization parameter (Q_p) are supported by the standard. The quantization step is doubled in size for every increment of 6 in $Q_{p.}$

Entropy Coding

There are two methods of entropy coding in H.264/AVC: CAVLC and CABAC. The simpler entropy coding method is supported in all profiles and uses a single infinite extent codeword table for all syntax elements except the quantized transform coefficients. The same set of codewords is used for each syntax element, but the mapping of the codewords to decoded values is changed depending on the statistics associated with each element. For transmitting the quantized transform coefficient a more efficient method called CAVLC is used. In this method, one of the numbers of VLC tables is selected for each symbol, depending on already transmitted syntax elements.

A more complex method called CABAC is used to improve coding efficiency. It is based on binarization, context modelling and binary arithmetic coding. Firstly, each symbol is binarized (converted to binary code) and then the value of each bit of the binary code (bin) is arithmetically coded. To adapt the coding to no stationary symbol statistics, context modelling is used to select one of several possibility models for each bin, based on the statistics of previously coded symbols.

Frame/Field Adaptive Coding

H.264/AVC includes tools for handling the special properties of interlaced video, since the two fields that compose and interlaced frame are captured at different instances of time. H.264/AVC allows encoder to make any of the following decision when coding a frame:

- To combine the two fields together and to code them as one single coded frame (frame mode).
- To not combine the two fields and to code them as separate coded fields (field mode).
- To combine the two fields together and compress them as a single frame, but when coding the frame to split the pairs of two vertically adjacent macroblocks before coding them.

The choice between the three options can be made adaptively for each frame in a sequence. When the election is done between the two first options is known as PAFF coding. When a frame is coded as two fields, each field is partitioned into macroblocks and is coded in a manner very similar to a frame, but an alternative zigzag coefficient scan pattern is used (it is shown in Figure 2.17) and individual reference fields are selected for MC prediction.

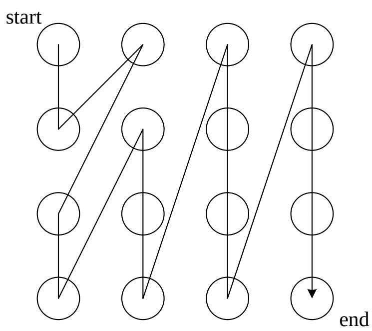
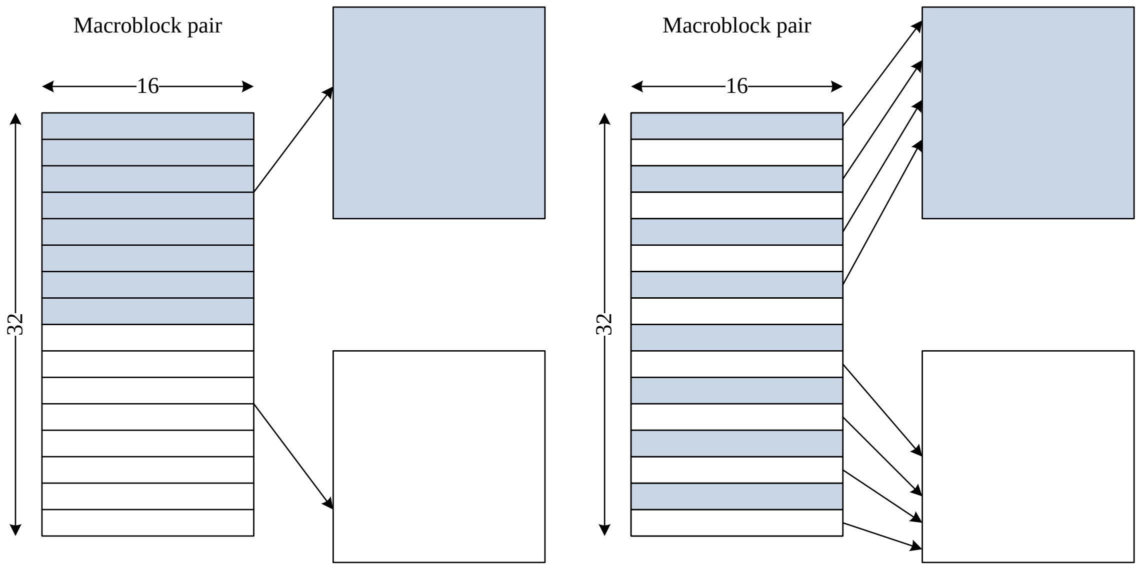


Figure 2.17. Reordering scan for 4x4 luma blocks

If a frame consists of mixed regions where some regions are moving and others are not, it is typically more efficient to code the non-moving regions in frame mode and the moving regions in the field mode. Therefore, the frame/field encoding decision can also be made independently for each vertical pair of macroblocks (16x32 luma region) in a frame. This coding option is referred to as MB-AFF coding. For a MB pair that is coded in frame mode, each MB contains frame lines and for a MB pair that is coded in field mode, the top MB contains top field lines and the bottom MB contains bottom field lines (see Figure 2.18).



(a) Frame mode (b) Field mode

Figure 2.18. MB adaptive frame/field coding

2.2.3 Profiles and Levels

H.264/AVC intends to be as generally applicable as possible. It has been development work a large range of applications, but different application means different requirements. In order to maximize interoperability while limiting the complexity, H.264/AVC defines profiles and levels.

A profile is defined as a subset of the coding tools that can be used to generate a conforming bitstream. A level is a specified set of imposed on values of the syntax elements in the bitstream. In H.264/AVC, the same levels definitions are used for all profiles defined. The combination between profile and level determinates the decoding capabilities since a decoder that satisfies a certain profile and level combination it has to be able to support all the tools and constraints defined in them.

In last version of H.264/AVC, there are twelve profiles defined (without SVC profiles). Three of them (Baseline, Main and Extended) were the profiles that appeared in first version published in May 2003, then in July 2004 the FRExt added four new profiles (High, High 10, High 4:2:2 and High 4:4:4). Since this amendment, other ones have been introduced leading to the current version of the standard.

As it has been said previously, each profile supports only a subset of the entire syntax of the standard and are designed to target specific applications area. A brief summary of these applications area is shown in Table 2.2. Regarding coding tools supported by these profiles, the key ones of the most used (Baseline, Main, Extended and High) are shown in Table 2.3.

Table 2.2. H.264/AVC Profiles and Their Applications

Profile name	Typical Applications
Constrained Baseline	Low-cost applications like video conferring and mobile applications.
Baseline	Low-cost applications that require additional error checking.
Main	Mainstream consumer broadcast and storage application.
Extended	Mainstream consumer broadcast and storage. Applications that require high compression and higher reliability.
High	High definition and Megapixel Broadcast and disc storage applications. HD DVD and Blu-ray Disc used it.
High 10	Professional applications that used interlaced video.
High 4:2:2	Professional applications that used interlaced video.
High 4:4:4 Predictive	Ultra high quality broadcast applications that demand lossless video.
High 10 Intra High 4:2:2 Intra High 4:4:4 Intra CAVLC 4:4:4 Intra	Production and contribution applications such as professional high definition video acquisition and edition.

If these supported coding tools are represented in a diagram as it shown in Figure 2.19, it can see that Extended Profile is a superset of the Baseline one and High Profile is a superset of the Main one. Furthermore, it can be appreciated that there are a set of common tools.

Regarding levels, sixteen are specified for each profile. Each level specifies upper limits for picture size, decoding processing rate, size of memory for multipicture buffers, video bitrate and video buffer size. The available levels are shown in Table 2.4. For more information regarding profiles and levels see Annex A of [20].

2.3 H.264/SVC

As it was mentioned previously, SVC is the scalable extension of H.264/AVC standard. It introduces a notion of layers within the encoded stream. A base layer encodes the lowest temporal, spatial, and quality representation of the video. Enhancement layers encode additional information that, using the base layer as a starting point, can be used to reconstruct higher quality, resolution, or temporal versions of the video during the decode process. In this way, a decoder can produce a video stream with certain desired characteristics by decoding the base layer and the number of the subsequent enhancement layers needed to achieve the desired result. During the encode process, a particular layer can be only encoded using reference to lower level layers. In this way, the encoded stream can be truncated at any point and still remain a valid, decodable stream.

Table 2.3. H.264/AVC profiles and supported functionalities

Functionality	Baseline	Main	Extended	High
I slices	X	X	X	X
P slices	X	X	X	X
B slices		X	X	X
SI and SP slices			X	
In-loop deblocking filter	X	X	X	X
CAVLC entropy decoding	X	X	X	X
CABAC entropy decoding		X		X
Weighted prediction		X	X	X
Field pictures		X	X	X
MB-AFF		X	X	X
Multiple slice groups (FMO)	X		X	
ASO	X		X	
RP	X		X	
DP			X	
Quantization scaling matrices				X
8x8 transform				X
8x8 Intra prediction				X

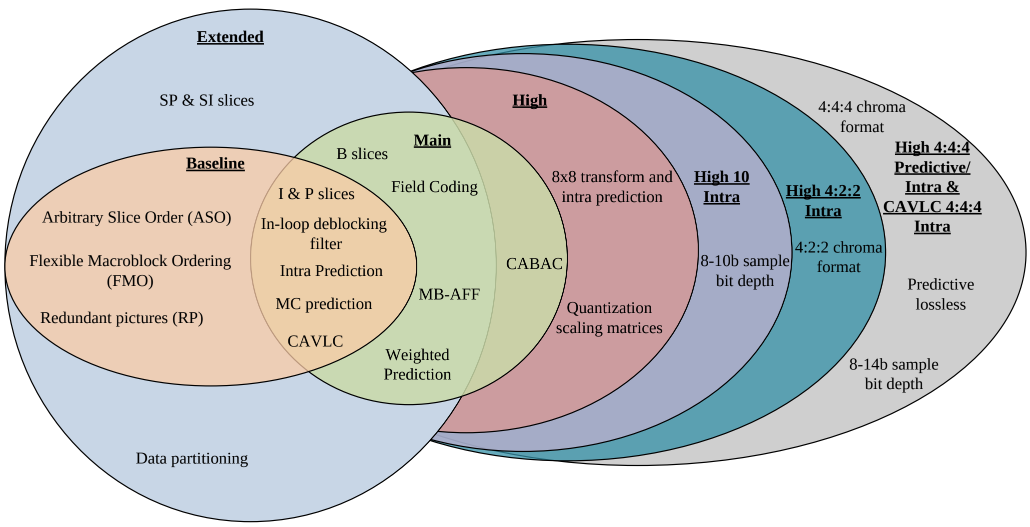


Figure 2.19. H.264/AVC profiles and tools

Table 2.4. H.264/AVC levels

Level	Max Picture Size and Format	Max Frame Rate (fps)	Max Bit Rate (bits/s)	Max Frame Size (MB)	Max Decoded Picture Buffer Size (MB)	Max Coded Picture Buffer Size (bits)
1	QCIF (176x144)	15	64 Kbps	99	396	175
1b	QCIF	15	128 Kbps	99	396	350
1.1	CIF (352x228)	7.5	192 Kbps	396	900	500
1.2	CIF	15.2	384 Kbps	396	2376	1000
1.3	CIF	30	768 Kbps	396	2376	2000
2	CIF	30	2 Mbps	396	2376	2000
2.1	625 HHR (352x576)	25	4 Mbps	792	4752	4000
2.2	625 SD (720x576)	12.5	4 Mbps	1620	8100	4000
3	625 SD	25	10 Mbps	1620	8100	10000
3.1	720p HD (1280x720)	30	14 Mbps	3600	18000	14000
3.2	SXGA (1280x1024)	42.2	20 Mbps	5120	20480	20000
4	2Kx1K (2048x1024)	30	25 Mbps	8192	32768	25000
4.1	2Kx1K	30	50 Mbps	8192	32768	62500
4.2	2Kx1080 (2048x1080)	60	50 Mbps	8704	34816	62500
5	3672x1536 (2.39:1)	26.7	135 Mbps	22080	110400	135000
5.1	4096x2304 (16:9)	26.7	240 Mbps	36864	184320	240000

Before SVC, other standard proposed its scalable version, but they have hardly been used due to with spatial and quality scalability, decoder complexity increases and coding efficiency decreases. In contrast to that, SVC achieves the same compression efficiency as H.264/AVC and the increase in decoding is moderate.

Similar to H.264/AVC standard, SVC is divided into a VCL and a NAL. This structure and a brief explanation can be found in H.264/AVC section. Scalability is provided at the bitstream level.

A bitstream with reduced spatial and/or temporal resolution and/or quality can be obtained by discarding NAL units from a scalable bitstream. The NAL units which are required for decoding of a specific spatial-temporal resolution and bit rate are identified by syntax elements inside the NAL header on by a preceding prefix NAL unit.

2.3.1 Network Adaptation Layer

As for H.264/AVC, the NAL is specified to formats the VCL representation of the video and provides header information appropriately for transportation by a variety of transport layers or storage media. Everything explained for H.264/AVC NAL in section 2.2.1 applies.

In contrast to the non-scalable profiles of H.264/AVC, the NAL concept of SVC was extended to provide a mechanism for bitstream manipulation and association of NAL units to scalable layers. The one-byte NAL unit header of H.264/AVC is extended by additional three bytes for the SVC NAL unit types. This extended header includes the parameters required for identifying the scalable layer which a VCL NAL units belongs to and includes additional information to help bitstream adaption as well. Each SVC bitstream includes a substream which corresponds with a non scalable profile of H.264/AVC [23].

2.3.2 SVC Video Coding Layer

A simplified block diagram of the SVC coding structure is shown in Figure 2.20. Each representation of the video source with a particular spatial resolution and quality that is included in a SVC bitstream is referred to as a layer and is characterized by a layer identifier. In each access unit, the layers are encoded in increasing order of layer identifiers. For the coding of a layer, already transmitted data of another layer with a smaller layer identifier can be employed. The layer to predict from can be selected on an access unit basis and is referred to as the reference layer. The layer with a layer identifier equal to 0 is coded fulfilling the requirements of one non-scalable H.264/AVC profile and is referred to as base layer.

The layers that employ data of other layers for coding are referred to as enhancement layers. The number of layers present in a SVC bitstream is dependent on the needs of an application. SVC supports up to 128 layers in a bitstream. With the currently specified profiles, the maximum number of enhancement layer in a bitstream is limited to 47 and at most 2 of those can represent spatial enhancement layers.

Similarly to H.264/AVC, the input pictures of each spatial or quality layer are divided into macroblocks and slices. The macroblocks are organized in slices, which can be parsed independently. For intra prediction, motion compensated prediction and transform coding a MB can be divided into smaller partitions or blocks.

Inside each layer, SVC follows the design of H.264/AVC standard for coding a single layer. The samples of each MB are predicted by intra or interprediction.

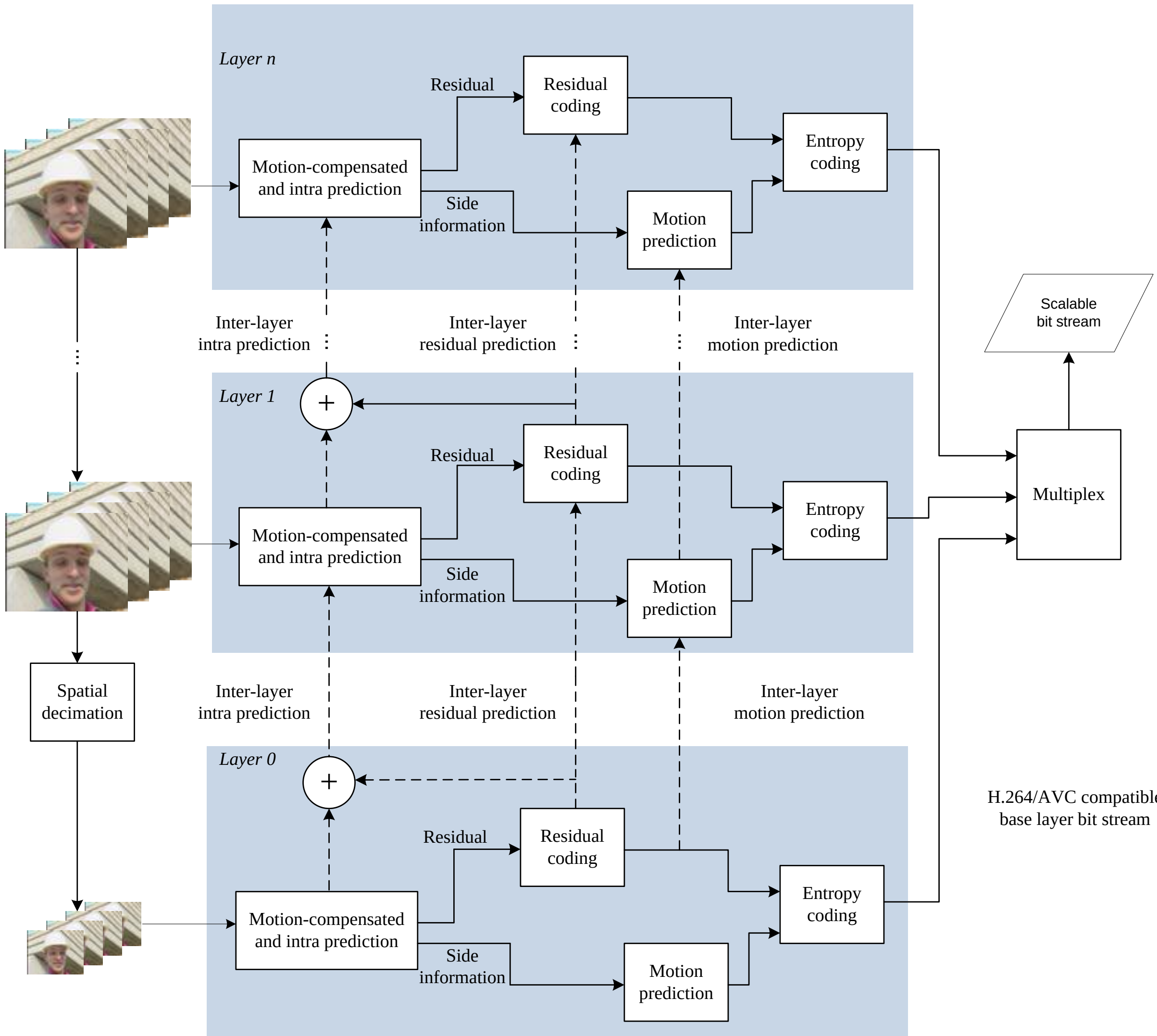


Figure 2.20. Basic coding structure for SVC for a MB

With intra prediction each sample of a block is predicted using spatial neighbouring samples of previously coded blocks in the same picture. With interprediction the prediction signal of a partition is build by spatial displaced region of previously coded picture of the same layer. The residual representing the difference between the original and the prediction signal for a block is transformed using an integer DCT based transform. The transform coefficients are scaled and quantized. The quantized transform coefficients are entropy coded together with other information such as MB coding type, the quantization step size and the coded intra prediction modes or the motion information consisting of identifiers specifying the employed reference pictures and corresponding displacement vectors. The MV components are coded using MVs of neighbouring blocks as predictors. The decoded representation of the residual is obtained by inverse scaling and inverse transformation of the quantized transform coefficients. The obtained decoded residual is then added to the prediction signal and the result is additionally processed by deblocking filter before output and storage as a reference picture for interprediction coding of following pictures.

In addition to these coding tools of H.264/AVC, SVC provides inter-layer prediction methods which allow an exploitation of the statistical dependencies between different layers for improving the coding efficiency of enhancement layers. All inter-layer prediction tools can be chosen on a MB or block allowing an encoder to select the coding mode that gives the highest coding efficiency.

As an important feature of SVC, each spatial and quality enhancement layer can be decoded with a single MC loop. For the employed reference layers, only the intra coded macroblocks and residual blocks that are used for inter layer prediction need to be reconstructed and the MVs need to be decoded. The computationally complex operations of motion-compensated prediction and deblocking only need to be performed for the target layer to be displayed.

Temporal scalability can be achieved by partitioning the access units into a temporal base and one or more temporal enhancement layers and restricting the encoding structure in a way that for each access unit of a specific temporal layer, only access units of the same or coarser temporal layer are employed for inter picture prediction. In the following subsections, the news features introduced by SVC (temporal, spatial, quality and combined scalability) are explained with more detail. For more information, see [23].

Temporal Scalability

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

In H.264/AVC and by extension in SVC, any picture can be marked as reference picture and used for motion-compensated prediction of following pictures. This feature allows coding of picture sequences with arbitrary temporal dependencies. In this way, to achieve temporal scalability, SVC links its reference and predicted frames using hierarchical prediction structures [24] 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 GOP. As it was mentioned previously, the temporal base layer represents the lowest frame rate that can be increasing by adding pictures of the enhancement layers.

There are different structures for enabling temporal scalability. One of these structures with a GOP of 8 (I7BP pattern) and therefore four temporal layers, is illustrated in Figure 2.21 where the temporal base layer is represented by TL0 and the successive temporal layers increase the identifier by 1. This structure provides another three independently decodable sub-sequences with 1/8, 1/4, and 1/2 of the full original frame rate.

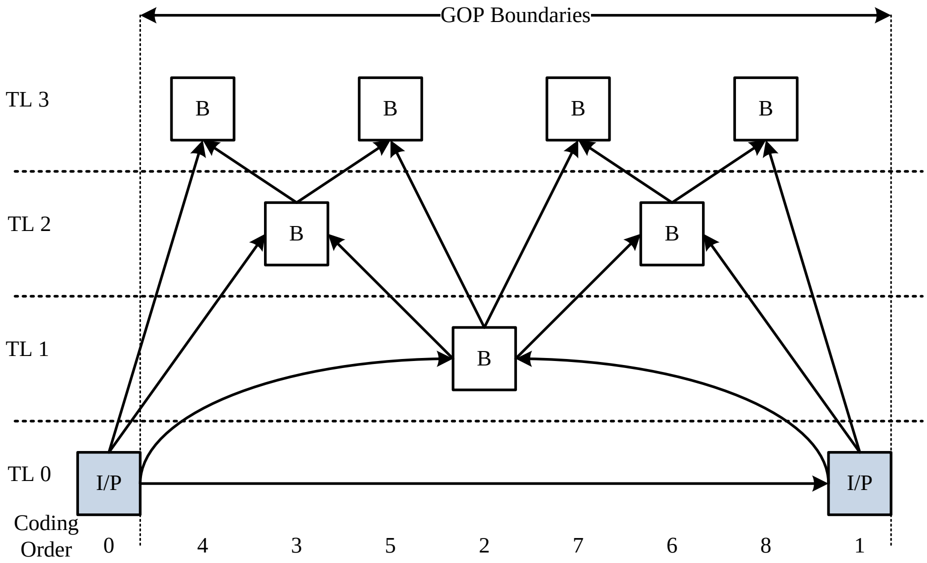


Figure 2.21. Hierarchical dyadic coding structure containing four temporal levels (*TL*)

Spatial Scalability

For supporting spatial scalable coding, SVC follows the approach of multilayer coding. Each layer corresponds to a supported spatial resolution and is referred to by a spatial layer or dependency identifier D (D=0 for base layer and increases it from one layer to another). The layers are coding following an oversample pyramid for each resolution (QCIF, CIF, 4CIF, 16CIF). A multi-layer structure that enables spatial scalability is shown in Figure 2.22.

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

SVC provides inter layer prediction methods which allow an exploitation of the statistical dependencies between different layers for improving the coding efficiency of enhancement layers. All these methods can be chosen on a MB or block allowing the encoder to select the coding mode that gives the highest coding efficiency.

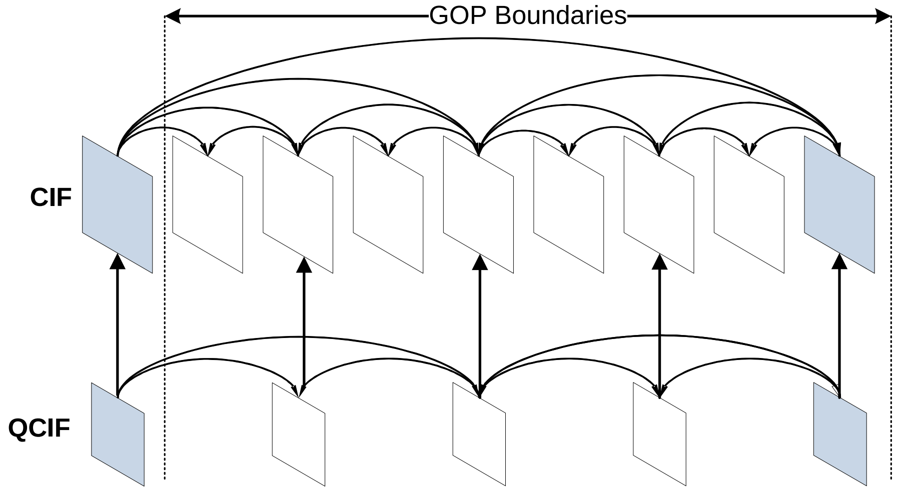


Figure 2.22. Multilayer structure with addional inter layer prediction for enabling spatial scalable coding

- *Inter Layer motion Prediction:* For spatial enhancement layers, SVC includes an additional MB mode that is signalled by *base mode flag*. The MB partitioning is obtained by upsampling the partitioning of the co-located 8x8 block in the lower resolution layer. The reference picture indexes are copied from the co-located base layer blocks and the associated MVs are scaled by factor of 2. These scaled MVs are either used unmodified or refined by an additional quarter-sample MV refinement. Additionally, a scaled MV of the lower resolution can be used as MV predictor for the conventional MB modes.
- *Inter Layer residual Prediction:* The usage of inter layer residual prediction is signalled by a flag (*residual prediction flag*) that is added to the MB syntax for spatial enhancement layers. When this residual prediction flag is set to 1, the upsampled residual of the co-located 8x8 reference layer blocks is subtracted from the enhancement layer residual (different between the original and the inter picture prediction signal) and the resulting different is encoded using transform coding.
- *Inter Layer intra Prediction:* When an enhancement layer MB is coded using the new MB mode and the co-located 8x8 block in its reference layer is intra coded, the prediction signal of the enhancement layer MB is built by using inter layer intra prediction for which the corresponding reconstructed intra signal of the reference layer is upsampled.

Quality Scalability

Quality scalability in SVC (SNR scalability) is intended to provide different levels of quality to the original video. It can be seen as a case of spatial scalability where the base and enhancement layers have identical pictures sizes, but different qualities. Different techniques for quality scalability have been designed: CGS, MGS and FGS. A specific design for FGS was not included in the definitive version of SVC. In CGS (Figure 2.23), quality scalability is obtained by varying quantization in the different layers.

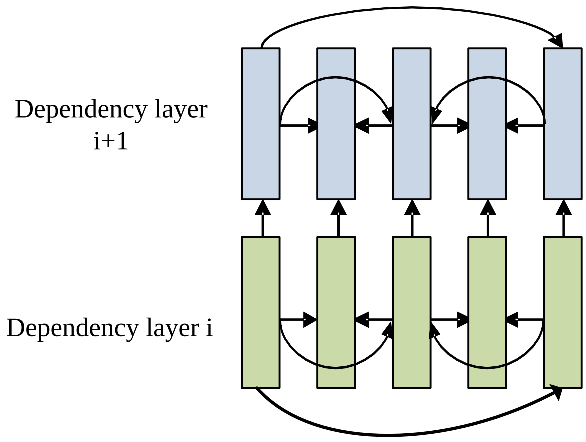


Figure 2.23. Dependencies for CGS quality scalability based on dependency layers

Firstly, the base layer, which is compatible with H.264/AVC, is generated. Then, for every quality layer added, the encoding techniques of H.264/AVC are combined with inter-layer prediction tools. This can make the compression of enhancement layers more efficient. MGS (Figure 2.24) uses techniques similar to CGS, but provides more flexibility. It is obtained by dividing the quantized coefficients over different packets. MGS allows switching between different MGS layers and the key picture concept, which allows the adjustment of a suitable trade-off between drift and enhancement layer coding efficiency for hierarchical prediction structures.

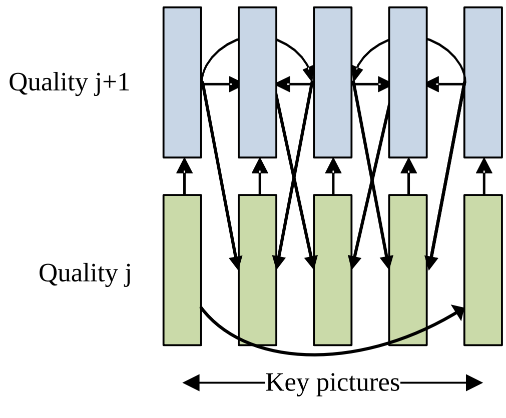


Figure 2.24. Dependencies for MGS quality scalability based on dependency layers

Combined Scalability

In the SVC extension of H.264/AVC, the basic concepts for temporal, spatial, and quality scalability can be combined as it is shown in Figure 2.25. However, a SVC bitstream does not need to provide all types of scalability.

As described previously, temporal scalability can be achieved by partitioning the access units into a temporal base and enhancement layers (identifier T) and restricting the encoding structure for each access unit of a specific temporal layer so that only access units of the same or a coarser temporal layer are employed for interprediction.

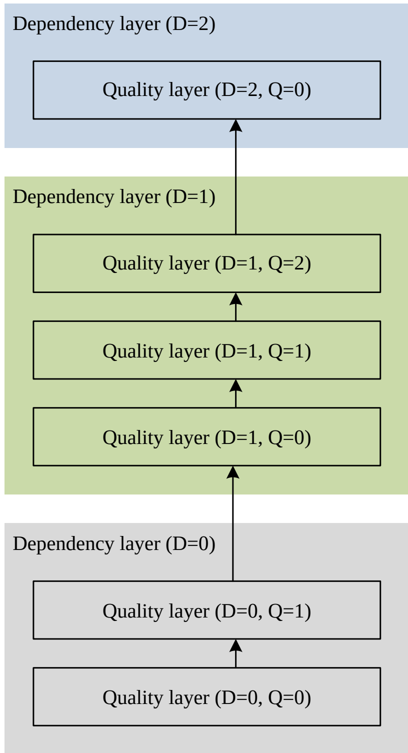


Figure 2.25. Example for structure of a SVC access unit

2.3.3 SVC Profiles and Levels

As in H.264/AVC, a profile is defined as a subset of the coding tools that can be used to generate a conforming bitstream. A level is a specified set of imposed on values of the syntax elements in the bitstream. The same levels definitions are used for all profiles defined.

The combination between profile and level determinates the decoding capabilities since a decoder that satisfies a certain profile and level combination it has to be able to support all the tools and constraints defined in them.

The SVC Amendment of H.264/AVC specifies three profiles: Scalable Baseline Profile, Scalable High Profile and Scalable High Intra Profile which are defined as a combination of the H.264/AVC profiles for the base and tools that achieve the scalable extension.

As it has been said previously, each profile supports only a subset of the entire syntax of the standard and are designed to target specific applications area. A brief summary of these applications area is shown in Table 2.5.

Table 2.5. SVC profiles and their typical applications

Profile name	Typical Applications
Scalable Baseline	Low decoding complexity applications like mobile broadcast, conversational and surveillance applications
Scalable High	Broadcast, streaming, and storage applications.
Scalable High Intra	Mainly designed for professional applications.

The main characteristics of these profiles are:

- *Scalable Baseline Profile*: The base layer conforms to the H.264/AVC Constrained Baseline Profile and enhancement layers supports B slices, weighted prediction, the CABAC entropy coding, and the 8×8 luma (CABAC and the 8×8 transform are only supported for certain levels). The support for spatial scalable coding is restricted to resolution ratios of 1.5 and 2 between successive spatial layers in both horizontal and vertical direction and MB-aligned cropping. Furthermore, the coding tools for interlaced sources are not included in this profile.
- *Scalable High Profile*: The base layer conforms to the H.264/AVC High Profile. The restrictions of the Scalable Baseline profile are removed and spatial scalable coding with arbitrary resolution ratios and cropping parameters is supported. Additionally, inter layer prediction is supported.
- *Scalable High Intra Profile*: It uses IDR pictures only. IDR pictures can be decoded without reference to previous frames. The base layer conforms to the H.264/AVC High Intra Profile with only IDR pictures allowed. All scalability tools are allowed as in Scalable High profile but only IDR pictures are permitted in any layer.

The profiles are represented in a graphical way as in H.264/AVC in Figure 2.26. Regarding levels, the same as specified for H.264/AVC are defined (see Section 2.2.3). Scalable bitstreams are categorized into a level as follows:

- A 2-layer scalable bitstream is within a level by:
  - Number of macroblocks of the enhancement layer.
  - Overall bit rate and buffer sizes.
- A 2+X-layer scalable bitstream:
  - Number of macroblocks:

- Let the layers be labelled as 0...2+X-1 with layer 0 being the base layer.
- Number of macroblocks of layer 2+X-1 plus number of macroblocks of layer 0...X 0.5.

- Overall bit rate and buffer sizes.

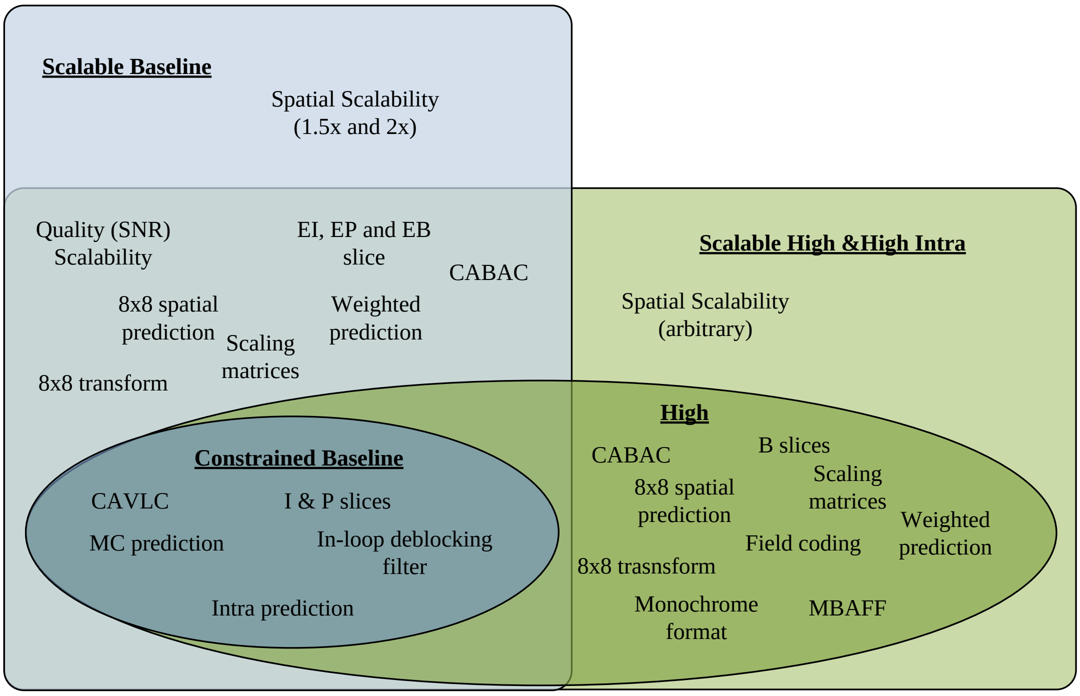


Figure 2.26. SVC profiles and tools

CHAPTER 3

VIDEO TRANSCODING ARCHITECTURES AND TECHNIQUES

Video transcoding is the process of converting a compressed video stream previously encoded with a determinate format or characteristics into another video stream encoded with a different codec or characteristics. The set of differences between the original and the transcoded video stream define the type of transcoding (see Figure 3.1).

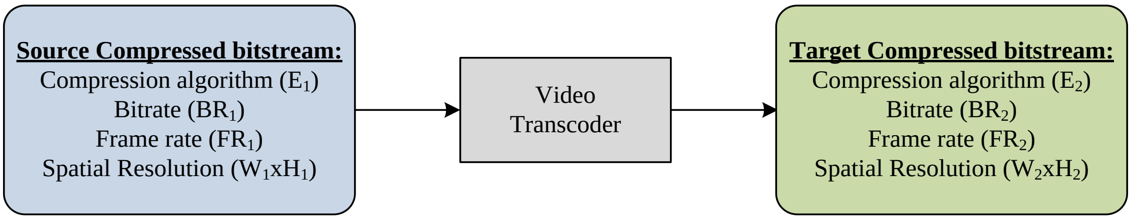


Figure 3.1. Video transcoding scheme

In case of the transcoding process translate the input video stream into a stream using the same standard as the original video sequence it is called *homogeneous transcoding*. Otherwise, if the video sequence is translated into a different format from the one used for encoding the input video sequence. This process is known as *heterogeneous transcoding*. The transcoding process should perform the conversion without making necessary the complete process of decoding and re-encoding.

There are several architectures and techniques used for transcoding video streams. In this chapter, an overview of them is done. Moreover, the state-of-the-art of H.264/AVC-to-SVC transcoding that is the main goal of this thesis is included.

3.1 Video Transcoding Architectures

The simplest transcoding architecture consists of a decoder followed by an encoder as is shown in Figure 3.2. In this case, the transcoding is performed by fully decoding the input video sequence and re-encoding the video sequence with the new desired characteristics. This simple architecture allows a transcoding without significant distortion in the image quality, so it can be used as reference transcoder for comparison of the performance of other architectures. Since the video is fully decoded and encoded again is very inefficient, because all the mechanisms involved in the encoding process have to be perform from the beginning and available useful information is not reused. This video transcoding architecture is known as *Cascade Transcoder*, *Reference Transcoding Architecture* or CPDT.

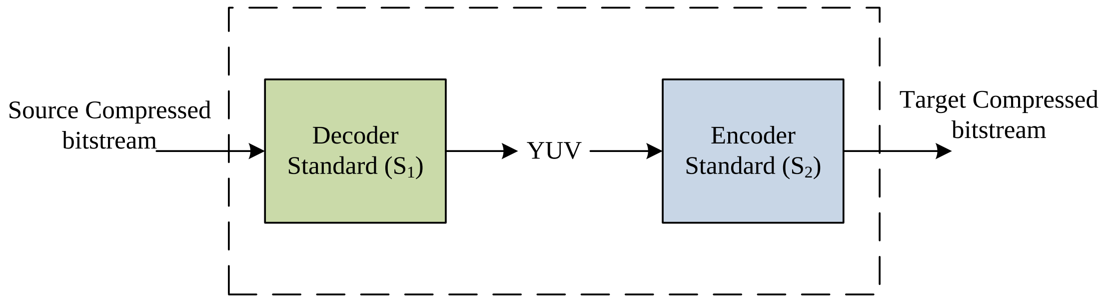


Figure 3.2. Cascade transcoder

In order to design an efficient video transcoder there are two key points that must be solved.

- Complexity reduction: The designed transcoder must reduce the coding complexity compared to the reference transcoder depicted previously. For that purpose, a possible solution could be to reuse information from the decoding stage of the transcoder for reducing the coding complexity of the most time consuming tasks of the encoding stage.
- Maximizing quality: While complexity is reduced, the quality performance of the process must be kept compared to the reference transcoding architectures.

The main goal of an efficient video transcoder taking into account the points mentioned above is to be capable of adapting in the shortest time possible the incoming video stream in the other with the desired characteristics while ensuring quality.

In the following sections, the most relevant transcoding architectures are described briefly. More information regarding this topic can be found in [25][26][27].

3.1.1 Open-Loop Transcoder

In this type of transcoder the video stream is partially decoded to extract the quantised DCT coefficients, MB type and residual parameters. As it avoids the ME, MC or transformations, the open-loop transcoders are computationally very efficient. A scheme of an open-loop transcoder is shown in Figure 3.3.

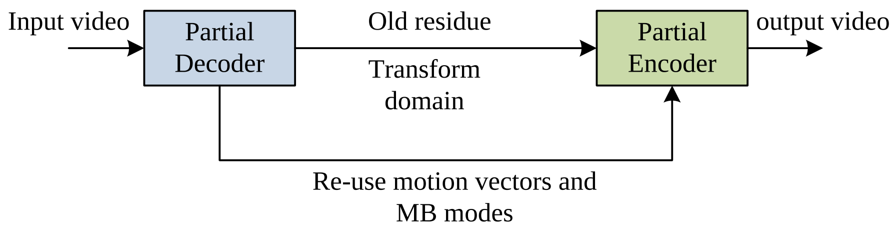


Figure 3.3. Scheme of an open-loop transcoder

Two types of open-loop transcoders are considered:

- With selective transmission: This type of transcoder (see Figure 3.4) discards some high frequency DCT coefficients.

Figure 3.4. Open-loop transcoder: high-frequency reduction

- With requantization: This type of transcoder (see Figure 3.5) performs an inverse quantization followed by a forward quantization with a coarser quantization parameter.

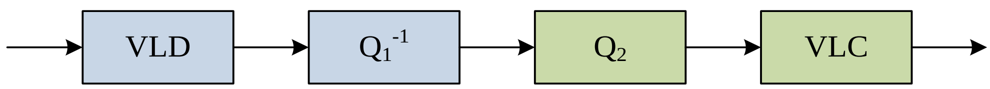


Figure 3.5. Open-loop transcoder: requantization

The main disadvantage of the open-loop transcoders is that an accumulative increment of the image distortion is produced. This phenomenon is called *Drift Error* [28] and is the result of the accumulated mismatch between the residual and the predicted image. In the literature there are different solutions for decreasing this visual degradation of the transcoded video sequence.

3.1.2 Closed-Loop Transcoder

This transcoding architecture was developed for overcoming the issue of the degradation of the video quality due to the drift error produced in the open-loop transcoders. This is possible by introducing a feedback loop to compensate the drift that results in an image quality significantly higher than in the open-loop transcoder, but in an increment of the computational complexity as well. A scheme of a closed-loop transcoder is shown in Figure 3.6

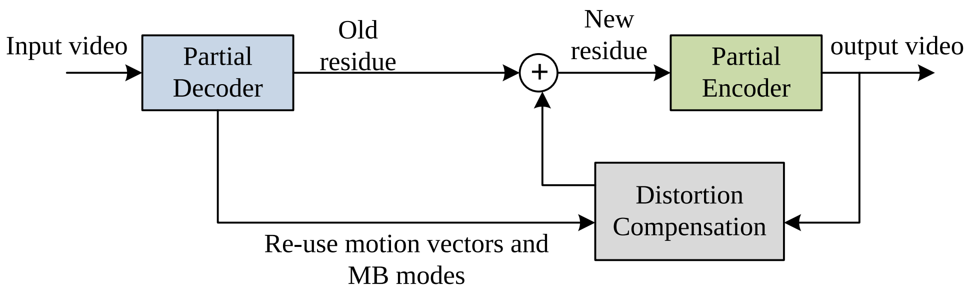


Figure 3.6. Scheme of a closed-loop transcoder

Closed-loop architectures can be used in different transcoding domains: spatial, frequency, hybrid, etc. All these transcoding domains are explained in the following subsections.

3.2 Video Transcoding Techniques

In this section different types of transcoders grouped by the problem that solve are presented. Moreover, the most representative examples reported in the literature are enumerated.

3.2.1 Bitrate Reduction

The bitrate reduction transcoders have been a popular research topic within the transcoding techniques. It is usual to find in the literature approaches that propose different techniques for converting from a high bitrate videos to a lower one. This process of transforming an encoded bitstream into another one with lower bitrate without changing video formats and while preserving the highest possible quality is called Transrating. This technique is very common solution for adapting a high quality bitstream to each end-user bandwidth constraints.

The transrating can be performed by using different techniques [29] as requantization (using a coarser quantisation step size) or by truncating DCT coefficients. In the first one, the quantization step is increased to match the target bitrate. This increasing of the quantization step results in a higher compression ratio caused by the decrease of the number of the representation levels of transformed coefficients. The second one is based in the irregular energy distribution of the DCT coefficients along the frequency domain. Since most of the energy of the coefficients is concentrated at the low frequency band, the high frequency coefficients have minimal impact on video quality. Figure 3.7 and

Figure 3.8 illustrate the transrating process using requantization or coefficients truncating respectively.

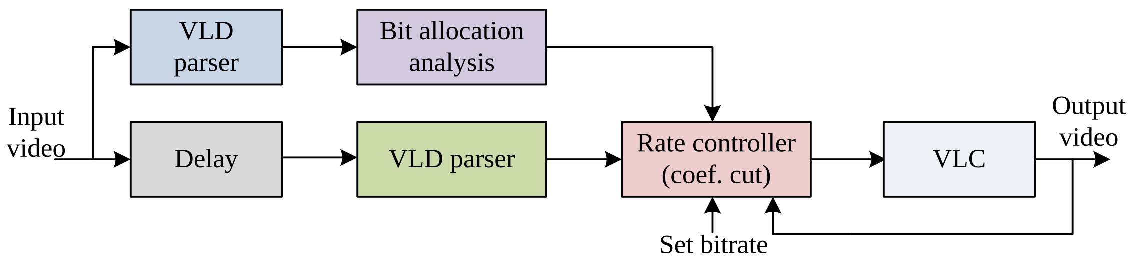


Figure 3.7. Transrating architecture: requantization

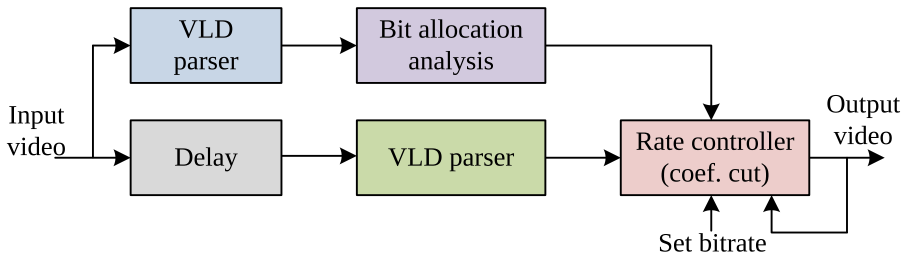


Figure 3.8. Transrating architecture: truncating coefficients

Some of the transrating proposals that can be found in the literature related to MPEG-2 and H.264/AVC are depicted below.

In 1996, Sun et al. evaluated transrating architectures [30] which reduce the bit rate of MPEG video streams. Two closed-loop architectures are presented which maintain a decoding and encoding loop, one that keeps the prediction decisions of the encoder and one that re-evaluates all prediction modes. These architectures are by definition drift-free and result in good visual quality. However, the complexity of these transrating architectures is high. Additionally, two open-loop architectures are proposed which discard high-frequency coefficients or requantize the residual data.

One of the most important proposals published in the field of transcoding was presented by Assunção and Ghanbari [31] in 1997. The authors describe the principles of operation of transcoding and the different architectures that may be proposed. Moreover, they propose a mechanism that adjusts the QP in function on the available bandwidth. Later, they proposed a transcoder in the DCT domain. Those proposals were used in the framework of MPEG-2 based transcoding.

In 1999 Werner et al. presented [32] a theoretical analysis of the requantization problem for MPEG-2 intra-coded pictures. Based on this analysis, efficient techniques for requantization are derived. The first method is based on distortion minimization of the requantization process which leads in general to higher bitrates. The second method is based on maximum a posteriori and results in a better rate-distortion performance.

Later, in 2006, Lefol et al, [33] used MPEG-2 requantization techniques for transrating of H.264/AVC video streams. They found that the properties of the video stream, in particular the number of intra-predicted macroblocks in motion-compensated pictures, have a major impact on the visual quality of the transrated H.264/AVC video streams.

In 2007, Shen et al. [34] presented a method for perfect requantization of H.264/AVC video streams. This technique, however, imposes constraints on the original video stream, and therefore complicates the requantization process.

In the same year, De Cock et al [35] investigated the requantization problem for intra-predicted macroblocks and proposed low-complexity compensation which preserves the visual quality of intra-predicted regions in H.264/AVC video streams. Afterwards, they made a hybrid architecture for transrating [36] which re-encodes the intra-predicted pictures and applies open-loop transrating or spatial/temporal compensation to motion-compensated pictures (depending on the picture type). This architecture has medium computational complexity and results in transrated video streams with acceptable visual quality.

In 2009, they presented a mixed transrating architecture [37] which combines different transrating techniques based on the picture/MB type. The mixed transrating architecture fully decodes and re-encodes intra-predicted pictures, while the motion-compensated pictures are transrated using open-loop requantization or compensation techniques. Applying spatial compensation in motion compensated pictures highly reduces the visual artefacts and therefore results in acceptable video streams with reduced bit rate. Adding temporal compensation in motion-compensated pictures further improves the visual quality, albeit to a smaller extent, but also increases the computational complexity and memory requirements of the transrating architecture.

In 2010, Lin et al. proposed [38] a dynamic rate control method for video transcoding for enhancing the visual quality in multipoint video conference. They also proposed a dynamic distortion weighting adjustment based on an H.263 homogeneous transcoder to improve the quality of the regions of interest such as faces on the conferees.

In 2011, Deknudt et al. proposed a complete transrating architecture [39] for H.264/AVC high-definition video bitstreams. Based on frequency-selective filtering, this architecture has a minimized complexity and is drift-error-free for intra-coded pictures.

3.2.2 Spatial Resolution Reduction

Spatial resolution reduction is necessary for adapting the multimedia contents to the different characteristics of end-user devices. Since, in general, the available multimedia contents are stored in high spatial resolution, exists a necessity of transcoders capable of adapting the spatial resolution to various scenarios.

Spatial resolution reduction is normally done at a 4:1 size reduction, where horizontal and vertical directions of the pictures are divided by 2. For example, in CIF to QCIF where every 4 blocks are converted into a single block. There are various techniques to perform this conversion.

One of these techniques is the MV mapping. As it was said previously, if every four MB are converted to one, the associate MVs need to be mapped. The number of MVs associated depends on the encoding standard and is a critical point because multiple MVs must be merged to a single one. The problem is depicted in Figure 3.9.

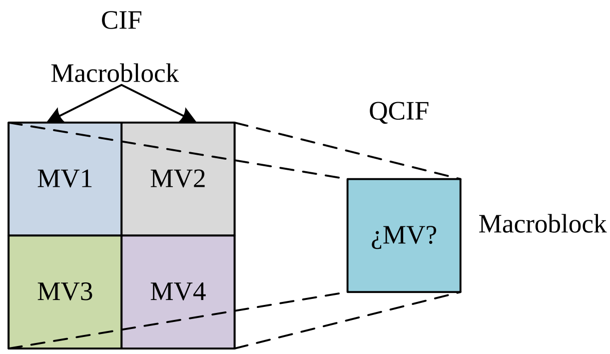


Figure 3.9. 4:1 MV mapping

There are some ways to solve this issue:

- Selecting one of the incoming MVs randomly.
- Taking the average or mean of some MVs.
- Taking the weighted average of the incoming MVs.
- Extracting the MV situated in the middle of the rest of the MVs by computing the Euclidean distances between each MV.
- Composing an MV by the corresponding MV with maximum DC coefficients of residual blocks in the source video.

In all of these methods, the magnitude of the MV is scaled down by a factor (normally by 2) to reflect the spatial resolution transcoding.

Another technique for spatial resolution reduction is the conversion of MB type. As said previously, in the spatial resolution reduction a group of four MB in the original video corresponds to one in the transcoded video. This scenario can lead to the issue of deciding MB mode. The problem is illustrated in Figure 3.10.

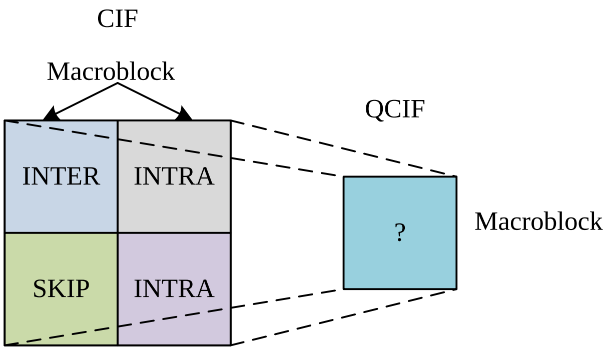


Figure 3.10. 4:1 conversion of MB type

Another way to reduce the spatial resolution is to only use the low frequency coefficients of the four original blocks to produce a new resize block. For conversion by a factor by 2, only the 4x4 DCT coefficients of each 8x8 block in a MB are retained. For that, the input coefficients are filtered by a set of DCT-domain filters.

Another common technique to reduce spatial resolution is filtering and subsampling. For example, a filter that can be used in both horizontal and vertical directions for luminance and chrominance and the image is then down-sampled by dropping every alternate pixel in both directions.

Different approaches based in the methods explained previously can be found in the literature. Some of them are depicted below.

In 1995, Tan and Ganbari [40] adopted a DCT decimation method in which every four input blocks of 8x8 pixels, corresponding to an area of 16x16 pixels are first DCT transformed.

In 1998, Bjork and Christopoulus [28] proposed a method to overcome the problem of conversion of MB type. In this method, the MB modes of the mixed MBs are all modified to intermode. The MVs for the intra-MB are reset to zero and so are corresponding DCT coefficients. In this way the input MBs that have converted are replicated with data from the corresponding blocks in the reference frame.

In 1999, Mohan et al. [41] proposed a solution to solve the issue of the conversion of MB type. If it exists at least one intra MB among the four MBs, the type selected will be intra. If there is not intra MB and at least one inter MB, then the type will be inter. If all the MBs are skip, then MB will be skip.

The same year, Shen et al. [42] proposed a general method that estimates the new MVs using the incoming ones from the sequence to transcode. To create the new MV, they take the average of mean of some MVs which are in the corresponding area in the source video or have the same direction.

Later, in 2000, Yin et al. [43] presented a technique where they first generate from the original compressed video an improved estimate of the MVs. Moreover, they proposed a compressed domain approach with data hiding to produce DCT residues by an open-loop method.

The same year, Shanableh and Ghanbari [44], in the framework of transcoding H.261/H.263 standards proposed a technique for lowing resolution. They extracted and composed a set of candidate MVs, from the incoming bitstream, to comply with the encoding format of the output bitstream. For the spatial resolution reduction they generated one MV out of a set of input MVs operating on the higher spatial resolution image. Finally, for the temporal resolution reduction they compose new MVs from the dropped frames MVs.

In 2002, Shen and Roy [45] proposed an optimized algorithm that avoids the reconstruction of original frames as much as possible. Moreover, the proposed algorithm took advantage of fast DCT domain down sampling methods as much as possible without the reconstruction of intra DCT version of original frames.Therefore, additional computational saving was achieved with negligible loss of quality. The algorithm was implemented in an MPEG video transcoding system.

Later, in 2003, Kong et al. [46] presented a new algorithm for MPEG-2 transcoding with spatial resolution reduction. The proposed method combined rate control and mode decision to achieve optimal transcoding performance using Lagrange multiplier algorithm.

In 2004, Yusuf et al proposed [47] a new MV composition technique by normalizing the low frequency DCT coefficients with the visual quantization matrix (VQM) derived from the study of human visual system (HVS).

In 2008, De Cock et al. [48] presented spatial resolution reduction transcoding architecture for H.264/AVC, which extends open-loop transcoding with a low-complexity compensation technique in the reduced-resolution domain. The proposed architecture removes visual artifacts from the transcoded sequence, while keeping complexity significantly lower than more traditional cascaded decoder-encoder architectures.

In 2010, Hongxing et al. [49] proposed an efficient scheme for reducing the spatial resolution of MPEG-4 video streams. The novelty of the scheme was to implement spatial resolution reduction by combining an elaborate transrating process with the subsampling at the receiver. By doing so, it can reuse the important information, such as MVs, MB modes.

Finally, in 2011, Bacquet et al proposed [50] a novel efficient downsizing video transcoder for H.264/AVC low bitrate compressed images. The algorithm accounted for new coding tools introduced by the H.264/AVC standard and applied directly on the transformed coefficients, thus reducing greatly the computational complexity of the transcoding algorithm.

3.2.3 Temporal Resolution Reduction

Reducing the frame rate may be needed to maintain a high quality of the encoded frames while saving bits or when the end-user equipment supports only a lower frame rate. Frame reduction enables the encoder to allocate more bits from the remaining video frames in the sequence, particularly those with high motion activity. Frame rate reduction involves dropping frames and, therefore, the incoming MVs are not valid because they point to the frames that do not exist in the transcoded bitstream, so it is necessary to derive a new set MVs taking into account the MVs of the dropped frames. The problem is illustrated in Figure 3.11 where frame n-1 is dropped, so a new MV to predict frame n from frame n-2 is estimated.

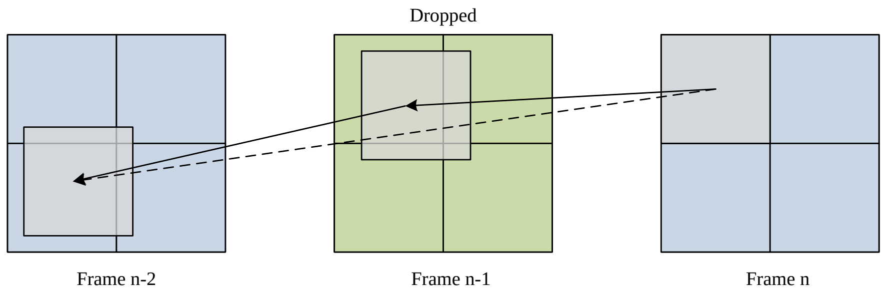


Figure 3.11. MV re-estimation

Various solutions to this problem have been proposed in the literature. Some of them are described in the following lines.

In 1998, Hwang and Wu [51] developed a bilinear interpolation method which estimated the MVs from the current frame by interpolation their values using MVs in the previous frames down until the previous non-skipped frames. The interpolated MV serves as the new search centre, and consequently reducing the search range significantly.

In 1999 Youn and Sun [52] presented a mechanism which selected the MV carried by a MB that has the largest overlapping segment with the block pointed by the incoming MV from the four neighbouring MBs. A frame is transcoded by taking the last preserved frame as reference.

In 2000, Shanableh [44] proposed a technique which accumulates all the MVs if the corresponding MBs of the dropped frames and add each resultant composed MV to its correspondence in the current frame.

In 2001, Fung et al. [53] proposed a frequency-domain approach to solve this problem. They developed an algorithm to adjust the number of dropped frame in a sequence depending on the information provided by the incoming MVs from the original encoded sequence. Re-estimation for MVs is done in the DCT domain using single add operations.

In 2002, Chen et al. [54] developed an algorithm which utilizes the activity of the MB to decide the choice of the MV. The activity is represented by the number of non-zero DCT coefficients.

In 2004, Lee et al. proposed [55] a temporal resolution reduction transcoding method that transformed an MPEG-4 video bitstream into an H.264 video bitstream. The block mode statistics and MVs in the MPEG-4 bitstream were utilized in the H.264 encoder for block mode conversion and MV interpolation methods. The proposed MV interpolation methods were developed not to perform brute-force ME again in the H.264.

In 2005, Shu et al. [56] introduced a new concept of motion change. Combined with motion activity, these two types of motion information were used to dynamically skip frames without seriously affecting our human perception. They also proposed a new frame-skipping control scheme with variable length prediction window.

Later, in 2006, Kwon et al. [57] presented a new rate-distortion optimized dynamic frame skipping transcoder for low bitrate video transmission in the compress domain in order to provide better visual quality for frame skipping transcoding while reducing computational complexity.

In 2007, Lee et al. [58] developed a temporal resolution reduction transcoding method that transformed an MPEG-4 video bitstream into an H.264/AVC video bitstream. The block modes and MVs in the MPEG-4 bitstream were utilized in the H.264/AVC encoder for the block mode conversion and MV interpolation methods. Four types of MV interpolation methods were proposed.

In 2008, Hsu et al. [59] proposed an effective two pass frame rate decision method for frame skipping video transcoding. It was used to estimate the dropped frame number to optimize frame number instead of the linear frame dropping used in the traditional frame

dropping transcoding. The proposed method decided the frame rate by using Lagrange optimization method. The Lagrange optimization was used to trade-off between the frame rate and bitrate for optimizing the frame complexity, frame rate and bitrate.
Later, in 2009, the same authors [60] proposed an arbitrary frame rate transcoding joint considering temporal and spatial complexity of frames in the adaptive length sliding window. The length of a sliding window was adjusted according to bandwidth variation in order to decide the number of skipped frames. The proposed method preserved significant frames and drops non-significant ones using the complexity measurements. Moreover, the proposed MV composition algorithm reduced the computations of ME process by adopting the coding feature of variable block sizes in H.264/AVC video transcoder.

3.2.4 Error-Resilience Transcoding

Transcoding can be used as well for enhancing the resilience of compressed video streams to transmission errors. For example, when transmitting video over wireless channels where the bandwidth is low and the error rate is higher than in wired channels becomes necessary this type of transcoding for accommodating the bitstream to the channel conditions and produce acceptable quality.

Figure 3.12 shows an example of error resilience transcoder with feedback. The transcoder extracts the video features from the incoming bitstream and estimates the client channel conditions according to the feedback channel information. This information is used to determine the error resilience policy.

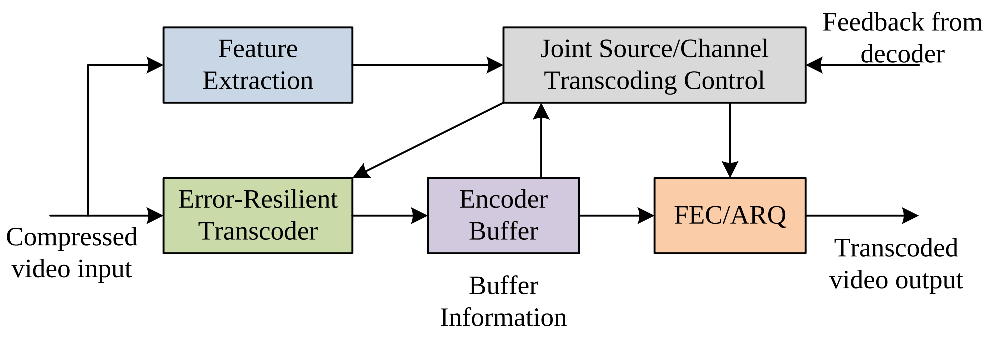


Figure 3.12. Framework of error-resilience video transcoder

For error-resilience purposes, the video transcoder can apply data partitioning and insertion of re-synchronisation markers into the incoming bitstream. Furthermore, the transcoder can apply unequal error protection to various segments of video data. Thus, the header data, followed by MVs, is assigned the highest protection, whereas the DCT coefficients are transcoded with a negligible level of protection. Moreover, the video proxy can forward the transcoded data in the form of separate sub-streams sent over different bearers in accordance with their sensitivity to errors and contribution to the

overall quality. This allows for the transmission of high priority error sensitive video data over high quality secure bearers for optimal video quality [61]. Some of the most relevant error resilience transcoders, proposed in the literature, are described below.
In 1996, Swann and Kingsbury [62] proposed an error-resilience MPEG-2 transcoding scheme. This technique proposed a transcoder divided into two parts, one between the encoder and the channel over the bitstream is transmitted and anther one between the channel and the decoder. Moreover, a mechanism for giving robustness based on error resilience entropy coding is proposed. The incoming bitstream is reordered and uses empty spaces let by shorter blocks.

Later, in 2000, De los Reyes et al. [63] presented a method that is built on three steps. First, they use a transcoder that enhances the encoded bitstream with spatial and temporal resilience. The transcoder increases the spatial resilience by reducing the number of blocks per slice, while temporal resilience is achieved by increasing the proportion of intra blocks transmitted in every frame. In order to maintain the same bitrate as the input bitstream, less significant coefficients are removed. In the second step, they created analytical models by characterizing how errors are propagated in a video. In the third step, they studied the output bitrate of different options being proposed for enhancing the robustness of the video sequences. As result of their analysis, they determine the best mechanism while maintaining the same bitrate as the original sequence.

In 2002, Dogan et al. [64] proposed an error resilience transcoder for GPRS mobile access networks. The transcoder serves as a gateway between different networks, adding to the encoded video sequence error resilience. The authors propose two mechanisms to enhance the robustness of the video: AIR or FCS. These mechanisms can work together or independently. Since the output bitrate is increased when the proposed mechanisms are used, the authors suggest changing the QP to avoid this increment.

Later, in 2004, Xia et al [65] presented in this paper a technique that differs from previous works by accounting for the inter-frame dependence in both video source requantization and error propagation of motion compensated video. Based on the rate-distortion models developed in this paper, an optimal Group-of-Picture based bit allocation scheme is proposed. We also propose a sub-optimal scheme that is suitable for a real-time implementation. Both the optimal and the suboptimal scheme achieve better PSNR performance than the fixed heuristic bit allocation scheme.

In 2007, Lie et al. [66] developed an error resilience transcoder based on the intra refresh technique for H.264/AVC videos. The algorithm focused on making decisions on the number of inserted intra MBs, allocations of proper MBs for intra refresh, and the quantization parameter for each P-frame, according to varying channel condition and frame contents.

In 2008, Shanableh et al. [67] proposed a number of video transcoding techniques for the purpose of adding error resiliency. The proposed solutions made use of distributed video coding technologies that were originally reported in the literature for distributing coding complexity between the encoder and the decoder. Three transcoding solutions were proposed: frequency-domain, time-domain and compressed-domain transcoding, and various decoding architectures were investigated. The proposed solutions served as a framework for boosting the error resiliency of pre-encoded video and can be applied to MPEG-2, and H.264/AVC coded streams.

In 2009, Chan et al. [68] addressed some issues on implementing error-resilient transcoding using the RPS. The proposed techniques classified the MBs of the requested frame into two categories. Then it selects the necessary MBs adaptively, processes them in the compressed domain and sends the processed MBs to the decoder.

In 2010, Zhang et al. [69] presented a scheme based on RPI to adaptively inject redundancy into bitstream compressed using H.264/AVC standard at media gateway. In this scheme, joint rate source-channel distortion metric and scalability metric are adopted to generate RPI which requires only several bits to represent for each picture when encoding a video sequence. The proposed scheme is able to utilize the system bandwidth efficiently while maintaining graceful quality degradation for all the clients in both online and offline applications.

In 2011, Zhou [70] developed an adaptive error-resilient scheme for temporal video transcoding in order to achieve good performances when temporally transcoded bitstream transmitted in the unreliable routines, After all original frames' motion intensity and error sensitivity values were calculated in the sliding window, every possible frame allocation's quality and error impact values due to the frame skipping were evaluated. Then, following the impact values, an optimum frame allocation selection approach based on the joint impact value was presented, which enables the dynamic balance between the coding efficiency and the error-resilient capability. Additionally, an effective intra refresh algorithm was advised to further enhance the robustness.

3.2.5 Format Transcoding

As it was said previously, if the video sequence is translated into a different format from the one used for encoding the input video sequence the process is known as heterogeneous transcoding. The heterogeneous video transcoding algorithms provide solutions for the incompatibility problem caused by the use of different video coding standards across different networking platforms. Therefore, the heterogeneous video transcoding involves video coding standard conversions for inter-network communications.

The problem of standard incompatibility has conventionally been overcome by employing a full decoder/re-encoder pair. However, this kind of cascade introduces a quality loss in video communications. On top of quality loss, decode/re-encode stage introduces a considerable amount of additional complexity resulting from the DCT/IDCT, MC and re-estimation processes. On the contrary, heterogeneous video transcoding is a straightforward algorithm which merely comprises video syntax conversions in the compressed domain. Therefore, the conversion algorithm consists of the following steps illustrated in Figure 3.13 [61]:

- Video frame header adjustment.
- Video data translation from one syntax to another
- Necessary bitstream stuffing for different synchronisation requirements of different standards.

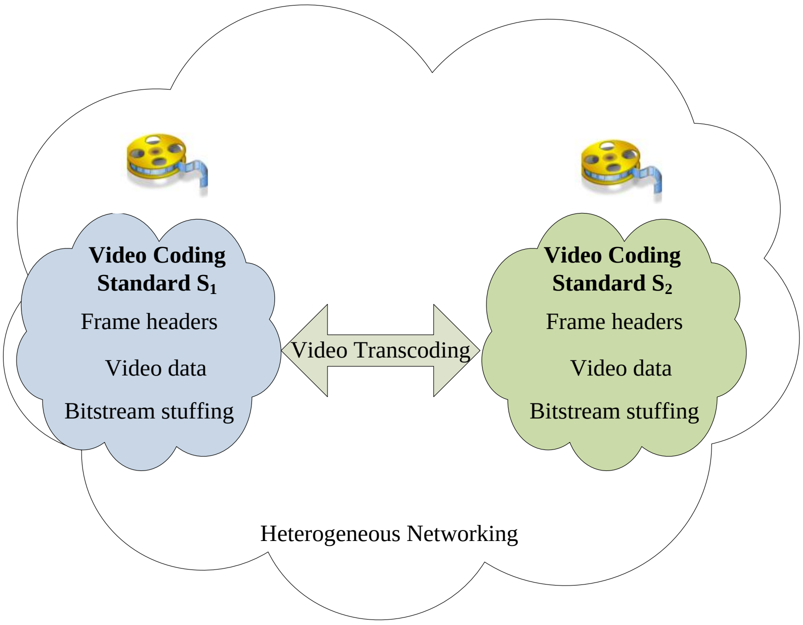


Figure 3.13. Inter-network heterogeneous video transcoding

Video data translation is the major process of the entire transcoding scheme. This process consists of enhanced mapping operations that involve transforming video parameters from one syntax to another one. However, this mapping process is still a much less complex and hence less time and power-consuming scheme than the full decode/re-encode technique. This is due to the fact that transcoding does not involve any computationally intensive transformations between the pixel and frequency domains, or any motion re-estimation and compensation processes. Moreover, syntax conversion does not require the inverse quantisation and re-quantisation of transform coefficients, except when bit rate reduction is also required. Consequently, when bit rate reduction is not an objective in the transcoding operation, the picture drift is avoided. However, when heterogeneous transcoding is combined with the homogeneous transcoding operation, the syntax conversion algorithm must be accompanied by one of the drift-free bit rate reduction schemes discussed in earlier sections of this chapter [61].

In the following lines, some approaches that can be found in the literature to try to solve one or more issue of this type of transcoding will be presented. The transcoding proposals related to H.264/AVC-to-SVC will be presented in section 3.3.

In 1999, Dogan et al. [64][71] presented a bidirectional transcoder between the H.263 and MPEG-4 video standards at low bitrates. Compatibility between different video standards can simply be achieved when the coders operate in their baseline profile. In this way, the difficulty of mapping some features of one particular standard, which are not supported by the other, is considerably simplified. To make easy the format conversion, encoded the sequence in one standard with coding tools that are not implemented in the other is avoid. Reducing the rest of the operation and directly mapping MVs, DCT coefficients, etc. the computational time is reduced.

The same year, Feamster and Wee [72] implemented a MPEG-2 to H.263 transcoder allowing the transcoding of an interlaced MPEG-2 bitstream to a lower bitrate progressive H.263 bitstream. It was a video transcoder to transmit digital TV (MPEG-2) in a wireless environment where the end-users devices decode H.263.

In 2000, Shanableh and Ghanbari [44] proposed a transcoder to convert MPEG-1 and MPEG-2 bitstreams to H.261 and H.263 bitstreams with lower spatial and temporal resolutions. They extracted and derived a set of candidate MVs from the incoming bitstream and operated with them to compose the necessary MVs for spatial and temporal resolution reduction.

In 2004, Bialkowski et al. [73] presented an H.263-to-H.264/AVC transcoder which focuses on intra macroblocks. This transcoder is based on the pattern similarities between the frequency domain prediction of H.263 and spatial prediction on H.264/AVC and uses the side information to simplify the mode and direction decision for intra prediction.

In 2008, Peixoto et al. [74] proposed a Wyner-Ziv-to-H.263 video transcoder. They proposed a mapping between the GOPs of the two standards and a ME refinement as well.

In 2008, Fernández-Escribano et al. [75] presented a MPEG-2-to-H.264/AVC transcoder based on the correlation between some information extracted from the MPEG-2 decoder stage and the H.264/AVC MB mode decision. In the same year, this idea was extended to implement an H.263 to H.264/AVC transcoder [76].

In 2009, Martínez et al. [77] presented a Wyner-Ziv-to-H.264/AVC transcoder that enables low cost video applications. This proposal accelerates the ME and mode decisions tasks of the H.264/AVC encoder stage of the transcoder by reducing the search area and narrowing down the MB types to be checked by the encoder according to a decision tree built previously.

In 2010, Tang et al. [78] proposed an efficient H.264/AVC block size partitioning prediction algorithm for MPEG-2-to-H.264/AVC transcoding applications. The algorithm used rate-distortion optimization techniques and predicted initial MVs to estimate block size partitioning for H.264/AVC. In addition to the fast block size partitioning algorithm, they also illustrated that using block size partitioning smaller than 8×8 (i.e., 8×4, 4×8, and 4×4) results in negligible compression improvements, and thus these sizes should be avoided in MPEG-2 to H.264/AVC transcoding.

In 2011, Xiaocong et al. [79] proposed a dedicated transcoder from AVS to H.264/AVC with reduced resolution, aiming to provide a fast and reliable solution for transcoding standard-definition videos to mobile contents. A multi-stage process was introduced for accurate motion and mode mapping.

In 2012, Corrales et al. [80] introduced an improved Wyner-Ziv to SVC transcoding framework to support homogeneous mobile video communications. Since Wyner-Ziv coding provides low cost video encoding, it is a suitable codec to encode video with less resources. On the other hand, the video delivery provided by SVC covers the needs of a wide range of homogeneous networks and different devices. As a consequence, Wyner-Ziv to SVC transcoding can offer a suitable framework to support scalable video communications between low-cost devices.

3.2.6 Hybrid Transcoding

There is a last group of transcoding techniques named hybrid transcoders. This kind of transcoders performs a combination of the transcoding techniques described previously.

Among the hybrid transcoders, stands out the multimedia traffic planning which has become a popular a popular topic of research due to the introduction and deployment of numerous diverse multimedia networks. One of the most significant characteristics of a network is its bandwidth, which controls the traffic load and determines the congestion. Among all the multimedia traffic types, coded video normally experiences the worst effects of congestion within a network. Congestion causes the decoded video to freeze for some time until the congestion is resolved. Once congestion has been solved and the streaming of video is resumed, the video encoder eventually skips all the missing video frames discarded by the network. The visual effect of this sudden gap is directly related to the duration of congestion and therefore the number of dropped video frames [61]. The

type of the discarded frames is important as well because the effect of discarding, for example, an inter frame is not the same as discarding an intra.
Due to the variety of possible situations and different types of devices a hybrid transcoder can be designed to meet the requirements of various links with different bandwidth characteristics. This transcoder would have a stack of video transcoders for the input bitstream producing several outputs at varying bitrates. Several buffers would be placed as well for controlling the transmission and to inform about the situation of the network. An example of a bank of transcoder is shown in Figure 3.14.

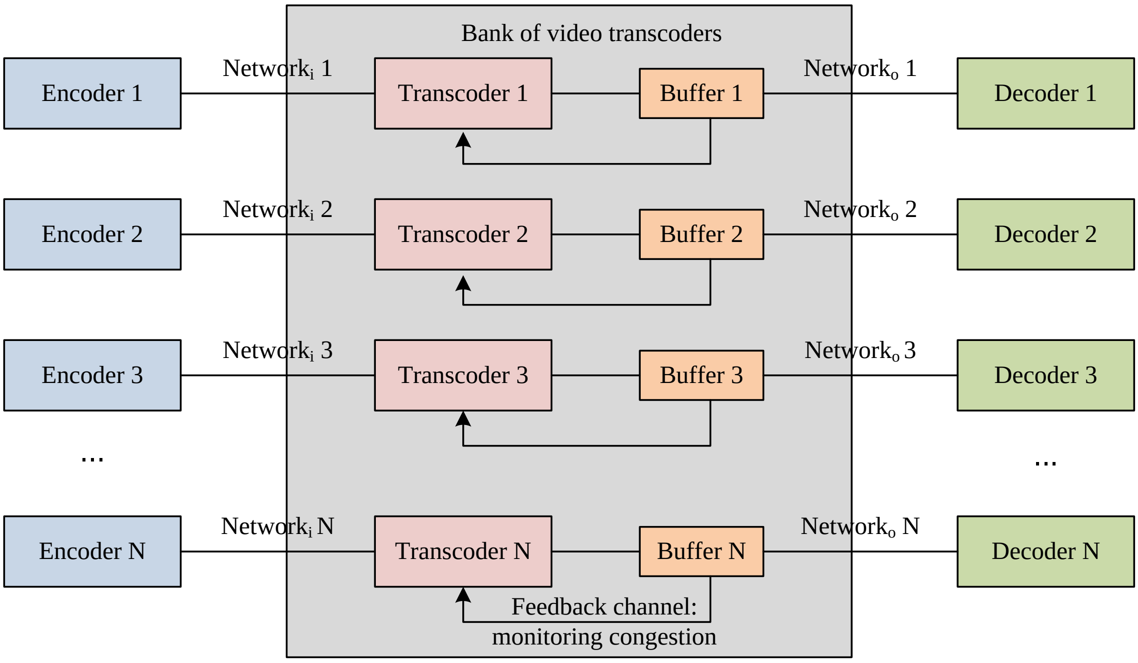


Figure 3.14. Stack of video transcoders

In the literature can be found some approaches regarding to this type of transcoders. Some of them are summarized in the following lines.

In 1998, Yeadon et al. [81] present results of integrating video encoded in H.263 into a heterogeneous mobile environment to provide multimedia support for the emergency services; the proposal focused on the features required to enable open working between a variety of applications, end-systems and networks and the performance of a very low bitrate encoder. An important result of the work is that they have demonstrated that in the majority of practical cases the quality of H.263 encoded video received by clients is governed by the performance of the end-system performing the compression which in a mobile environment is likely to be of low capability.

In 1999, Iannaccone et al. [82] proposes a set of techniques for the replication of the similar video data in each of the output streams may also be generated for multi-rate video transmission. Moreover, with the use of a feedback channel to report the changing

congestion conditions of a particular network, the video transcoder is able to dynamically adapt its output rate to the reported channel conditions.
Then, in 2001 Dogan et al. [83] proposed a video transcoding algorithm which transforms the input bitrates to suit channel bandwidth bottlenecks whilst also providing substantial amount of error resilience at the near required transmission rates. The simulation results also verify the necessity of the proposed error-resilient video transcoder. This particular architecture provides an adaptive resilience scheme which protects the error sensitive high motion data. The increase in the output bitrate is then compensated by the rate management feature of the video transcoder. The state of the network is monitored, and the system tries to give the maximum quality to the client.

In 2002, Martin [84] described the design and development of an adaptive environment for rendering of 3D models over networks. This environment monitored the resources available and selected the appropriate transmission and representation modalities to match these resources.

In 2004, Shin and Koh [85] presented an approach that determined the optimum time to apply transcoding by considering the potential benefits that can be realized. For instance, in order to save disk bandwidth for frequently accessed content, it pre-creates and stores multiple QoS versions. On the other hand, in order to save disk space for rarely accessed content, it stores only a single QoS version and performs transcoding on the fly.

3.3 State-of-the-art in H.264/AVC-to-SVC Transcoding

As it was mentioned previously, the scalable extension of H.264/AVC (SVC) was standardized in 2007, so transcoding proposals that involve this standard are recent. Different techniques for transcoding in this framework have been proposed. Most of the proposals are related to quality-SNR scalability, although there are several related to spatial and temporal scalability.

For quality-SNR scalability, the first one [86] was presented in 2006 and performs a transcoding from H.264/AVC to FGS streams. Although it was the first work in this type of transcoding, 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 2007, a transcoding approach from single layer H.264/AVC bitstream to SNR scalable SVC streams with CGS layers was presented by Jan de Cock et al. in [87].They proposed architecture for transcoding to SVC bitstream with two layers where depending on the slice and MB type, a distinction is made between spatial and temporal transform-domain compensation. Furthermore, two buffers are provided, one for requantization error values from current frame and another one for temporal compensation of inter-predicted macroblocks. This architecture can be seen in Figure 3.15.

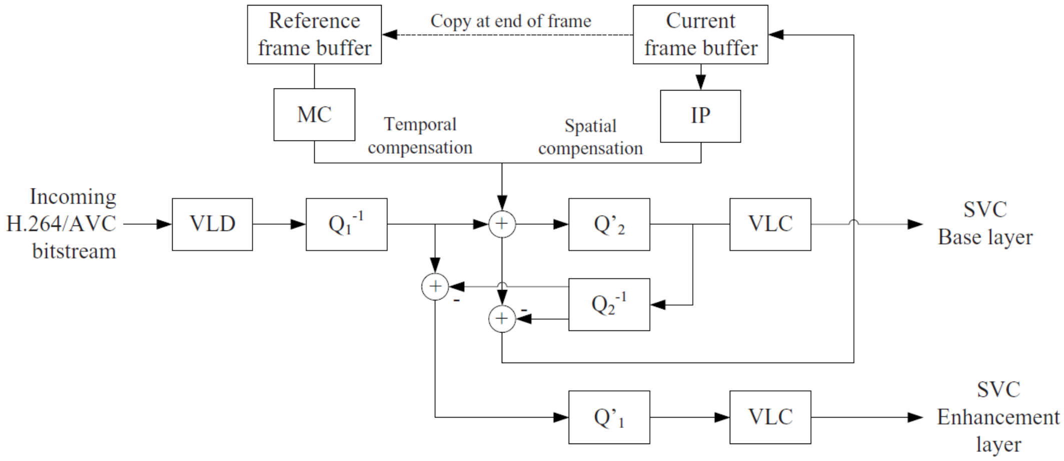


Figure 3.15. Drift-compensating transcoding architecture [87]

In 2008, Jan de Cock et al. [88] presented a proposal where the normative bitstream rewriting process implemented in SVC standard is used to reduce the computational complexity of H.264/AVC to SVC transcoding compared to [87]. It is based in combining the forward and inverse quantization processes, as shown in Figure 3.16.

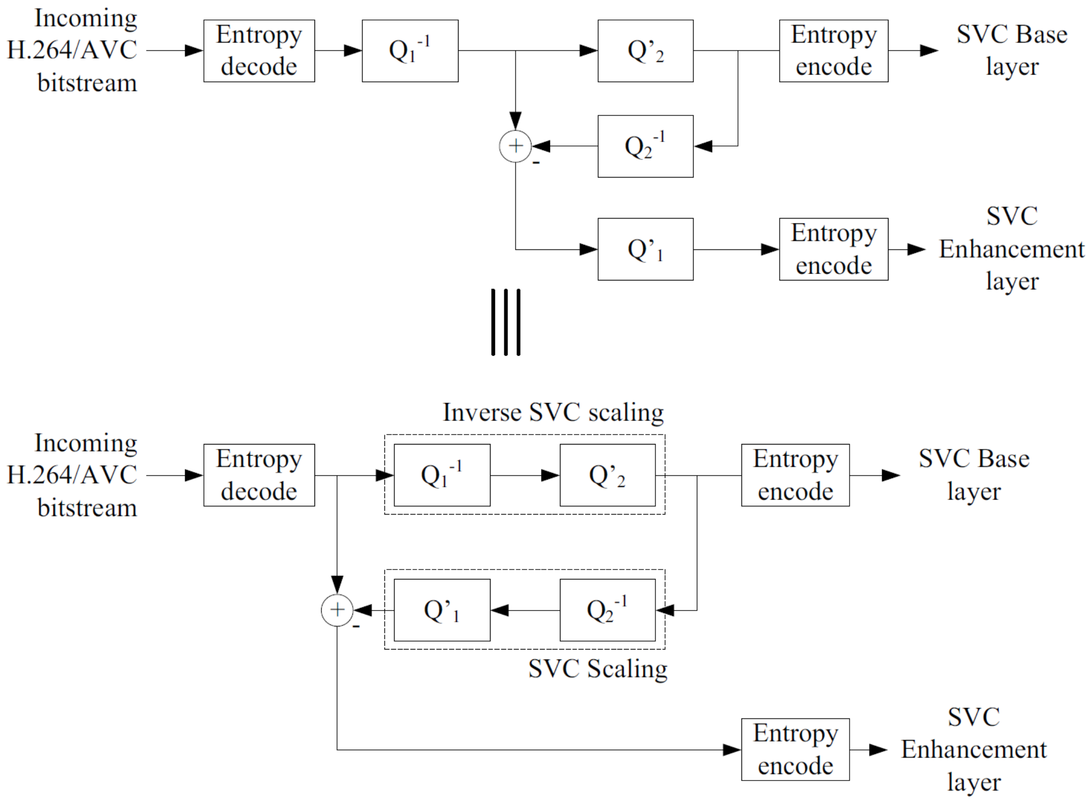
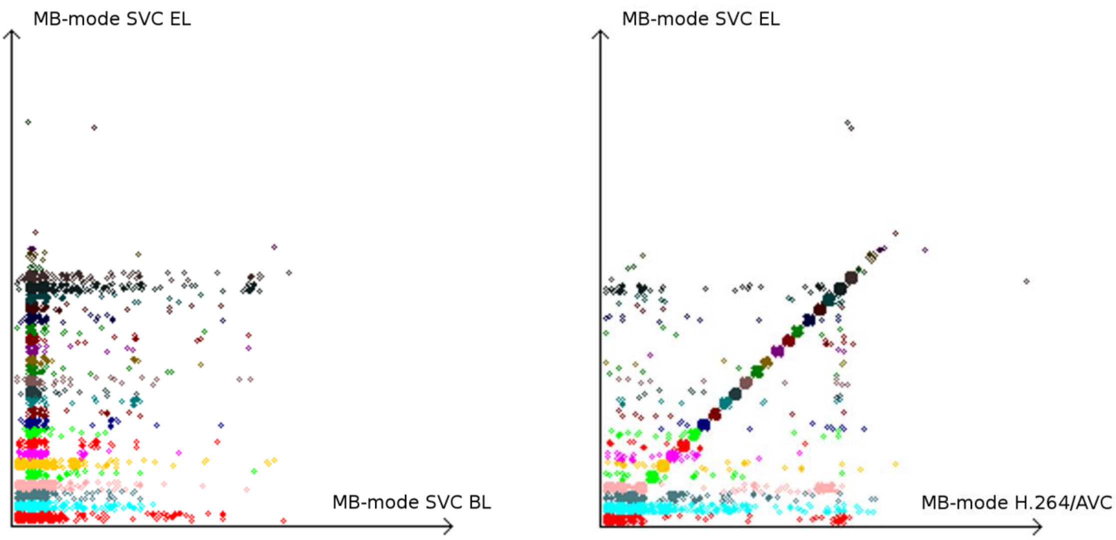


Figure 3.16. H.264/AVC-to-SVC rewriter [88]

Later, in 2009 Jan de Cock et al. [89] presented different open-loop architectures for transcoding from a single-layer H.264/AVC bitstream to SNR-scalable SVC streams with CGS layers.

In 2010, Van Wallendael et al. [90] proposed a simple closed-loop architecture that reduces the time of the mode decision process. This is done using two sources of information: the mode information from the input H.264/AVC video stream and the base layer of SVC that provides information for accelerating the encoding of the enhancement layers. This method is based in the relation between the modes of H.264/AVC, the base layer of SVC and the enhancement layers of SVC. In Figure 3.17 is shown this relationship.



(a) Relation between MB mode of BL and MB mode of EL

(b) Relation between MB mode of H.264/AVC and MB mode of EL

Figure 3.17. Relation between the MB modes [90]

Then, in 2011, Van Leuven et al. proposed two techniques to improve the previous proposals [91][92]. These methods are based in the same concepts as [90], but improve the results. The first one does it by exploiting more information from the input H.264/AVC bitstream and the second one by combining open- and closed-loop architectures. This combined architecture is shown in Figure 3.18.

For spatial scalability, a proposal was presented by Ravin Sachdeva et al. [93] in 2009. They presented an algorithm for converting a single layer H.264/AVC bitstream to a multi layer spatially scalable SVC video bitstream, containing layers of video with different spatial resolution. Using a full-decode full-encode algorithm as starting point, some modification are made to reuse information available after decoding a H.264/AVC bitstream for ME and refinement processes on the encoder. The scalability is achieved by an Information Downscaling Algorithm which use the top enhancement layer (this layer has the same resolution as the original video output) to produce different spatial layers of the output SVC bitstream.

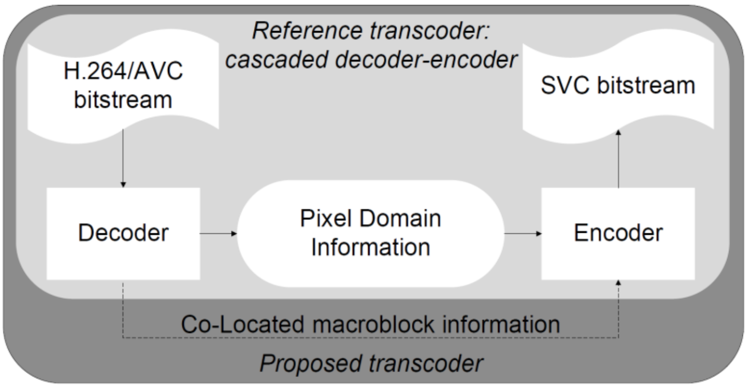


Figure 3.18. Overview of the proposed combined open- and closed loop architecture for H.264/AVC-to-SVC transcoding [92]

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 [94] 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 [95]. The method presented was applied in the Baseline Profile and reused information from the mode decision and ME processes from the H.264/AVC stream.

There are several more proposals concerning transcoding from H.264/AVC-to-SVC with temporal scalability which are part of this thesis [99][100][101][102][108][109][112]. These techniques were proposed in 2010 and 2011. The insights of the proposed approach are shown in the following chapters.

In the field of the reverse transcoding (SVC bitstream to H.264/AVC bitstream), in 2006 Segall proposes in [96] a technique which allows rewriting an SVC bitstream with multiple quality layers to a single layer H.264/AVC bitstream. A year later, in 2007, a normative bitstream rewriting process [97] which allows converting an SVC bitstream with multiple CGS layers into a single-layer H.264/AVC bitstream was added to the SVC specifications. In 2008, Peter Amon et al. presented a work [98] where they investigates SVC to H.264/AVC transcoding with emphasis on the conversion of SNR scalability layers in SVC to a single layer H.264/AVC.

CHAPTER 4

MOTION BASED H.264/AVC-TO-SVC TRANSCODING

In this chapter, the first proposal of this thesis is described. First of all, the motivation are explained, then the proposed technique is shown and, finally, a performance evaluation is done including an study of the time consuming in encoding every temporal layer and the impact of the GOP size and the number of temporal layers where the proposal is applied.

4.1 Observations and Motivation

ME process was explained in section 2.2.2, but a brief explanation is that this process consists in finding a region in a reference frame that matches as much as possible to the current MB. In order to find this region, a search area situated in the reference frame is defined. That search area is centred 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, 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. As a search over all the search area has to be done for every MB and sub-MB, this is one of the most time-consuming tasks carried out at the encoder stage.

In this chapter an improvement of the H.264/AVC-to-SVC transcoder focusing on reducing the time spent by this task is presented. The idea of this improvement consists on reusing the motion information that can be gathered in the H.264/AVC decoding algorithm (as part of the transcoder) to accelerate the SVC encoding process (also included in the transcoder). The MVs of the decoder give us an approximate idea of the movement of the scene and can be used to reduce the search area in the encoder part.

Adapting the search area based on the MVs of the incoming MB could reduce the time consumption of the transcoder without severely impacting the PSNR or the bitrate. To reduce the search area adaptively, an approach based on the length of the MVs extracted from the decoder part is developed. A scheme of the main idea of the approach is shown in Figure 4.1

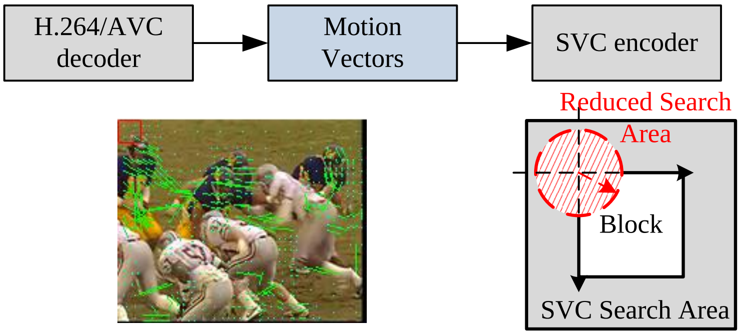


Figure 4.1. Scheme of the proposed transcoder.

Due to the different prediction structures used by H.264/AVC without temporal scalability and SVC, it can exist difference MB partitioning between H.264/AVC and SVC for each MB, and. therefore, a different number of MVs. Moreover, as the reference frame used by SVC can be different from the used by H.264/AVC, the information of the movement needs to be adjusted for building a search area more accurate to the real movement of the scene.

Finally, in SVC bitstreams with dyadic structures, the distribution of the frames within the temporal layers is not equitable and the encoder does not need the same time for encoding every temporal layer. This distribution can be seen in Figure 2.21 and is explained in section 2.3.2.

Both the proposal for reducing the time necessary for ME and the study of the time spends in every temporal layer are described in the next sections.

4.2 Dynamic Motion Estimation Window Approach

In this section, an approach based on varying the ME search area dynamically is presented. This technique can be used for different profiles and different GOP sizes of SVC as it is shown in the following subsections.

This section discusses the proposed dynamic ME search window algorithm for Baseline Profile [99], Main Profile [100] and the impact that has in the proposal the number of temporal layers [101] and the GOP size chosen [102].

4.2.1 First Stage: Constructing a Reduced Search Area

As it was said previously, the idea of ME consists of eliminating temporal redundancy by means of determining 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 ME process by reducing the search area dynamically and efficiently. This smaller search area is determined by the circumference centred at the point (0, 0) for each MB or sub-MB. This circumference has a radius which varies dynamically depending on the length of the corresponding MV in H.264/AVC. This idea is depicted in Figure 4.2.

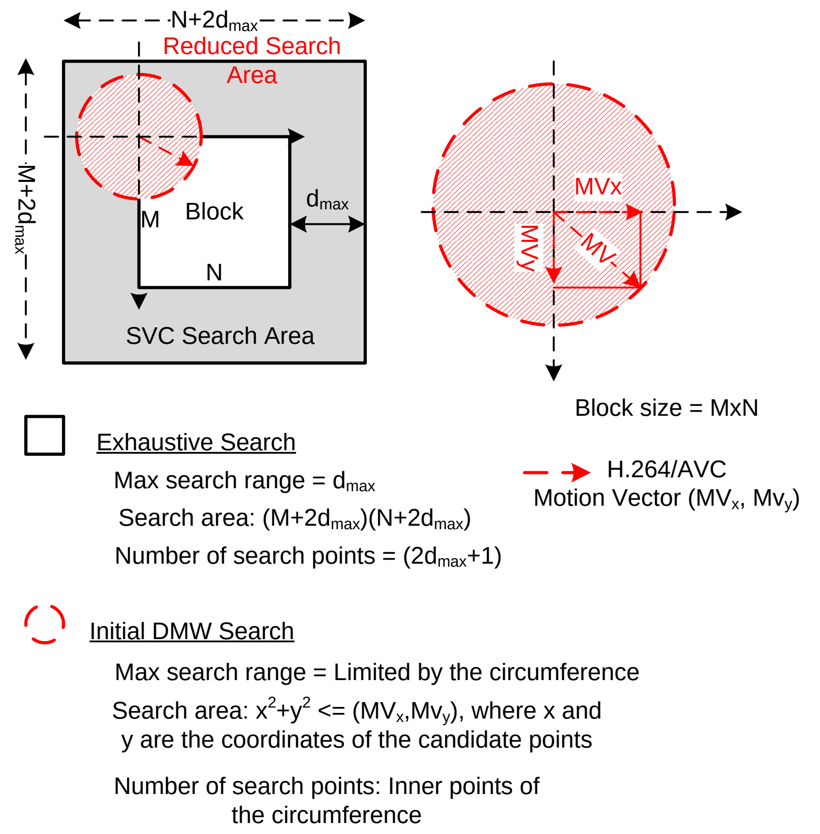


Figure 4.2. Proposed initial reduced search area.

The first problem to overcome is that there is not always a one-to-one mapping between previously calculated H.264/AVC MVs and the incoming SVC MVs. This is due to the fact that MB partitioning can be different for the same picture in H.264/AVC and SVC as illustrated in Figure 4.3. Also, there is a MV for each MB or sub-MB, so the number of

MVs associated to a MB in H.264/AVC can be different from the number of MVs associated with an MB in SVC

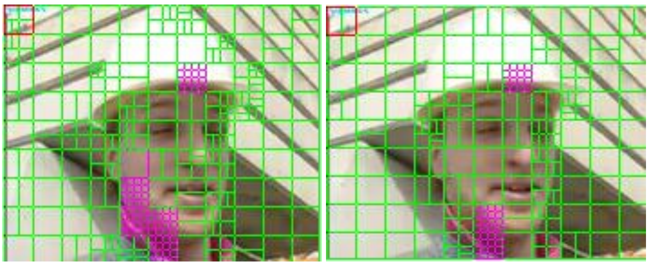


Figure 4.3. MB partitions generated by H.264/AVC (left) and SVC (right) for the 2nd frame in the Foreman sequence (QCIF).

The present approach tries to tackle this problem by using the average of the incoming MVs of the collocated H.264/AVC MB as shown in Figure 4.4.

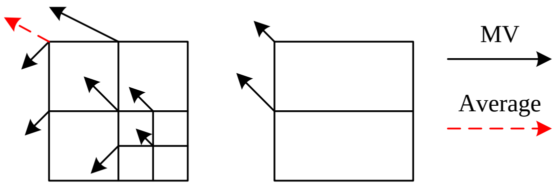


Figure 4.4. MB in H.264/AVC with its MVs and the matching MB in SVC with its corresponding MVs.

For adapting this proposal to be run in Main Profile, it must be taken into account that 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. An example of prediction modes in B frames is shown in Figure 4.5.

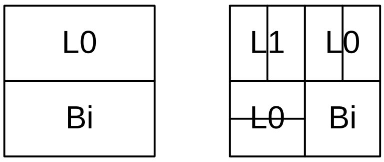


Figure 4.5. Examples of prediction modes in B macroblocks (L0: list0, L1: list1, Bi: bi-predictive)

Due to the difference in 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 from list0, but SVC needs the reference from list1 or vice versa or even a bidirectional prediction is performed that requires MVs from both lists. In these cases, the assumption is made that the length of the MV from both lists for an MB is the same.

The new search area will be defined as follows: let C be the circumference which restricts the search area with centre on the upper left corner of the MB or sub-MB defined as:

$$C^2 = r_x^2 + r_y^2 \tag{4.1}$$

where r_x and r_y depends on the average of the MVs of the H.264/AVC MB (MV_x and MV_y) and have a minimum value of 1 to avoid applying search ranges that are too small. The values are defined by:

$$r_x = \max(MV_x, 1),\ r_y = \max(MV_y, 1) \tag{4.2}$$

Let A be the search area used by SVC and (x,y) the coordinates to check. The new search window will be limited by the area S defined in:

$$S = (x, y) \mid (x, y) \in (A \cap C) \tag{4.3}$$

The idea consists of applying this reduced search area to all the MB partitions and sub-partitions checked by the SVC encoding algorithm. Thus, the MB mode decision algorithm in SVC is kept untouched.

4.2.2 Second Stage: Adjusting Reduced Search Area

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

Due the different GOP patterns between H.264/AVC (traditional IPPP in Baseline Profile and IBBP in Main Profile) and SVC (hierarchical), it is usual to have cases where MVs have been calculated using a reference frame, but this reference frame is not necessarily the same in the prediction structure of SVC, so the distance between the current frame of SVC and its reference frame can be different from that used in H.264/AVC. In general, hierarchical GOP structures will cause motion-compensated prediction to use a longer distance between a frame and its reference as is shown in Figure 2.21. This distance increases 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 Figure 4.6.

Therefore, the expression depicted in Equation 4.2 is multiplied by this correction factor, so r_x and r_y will be calculated as:

$$r_x = coef(n) \cdot \max(MV_x, 1),\ r_y = coef(n) \cdot \max(MV_y, 1) \tag{4.4}$$

Here, *coef* depends on the number of the temporal layer (n) where the frame is in and it is calculated as:

$$coef\,(n) = GOP_{length} \,/\, 2^{n} \tag{4.5}$$

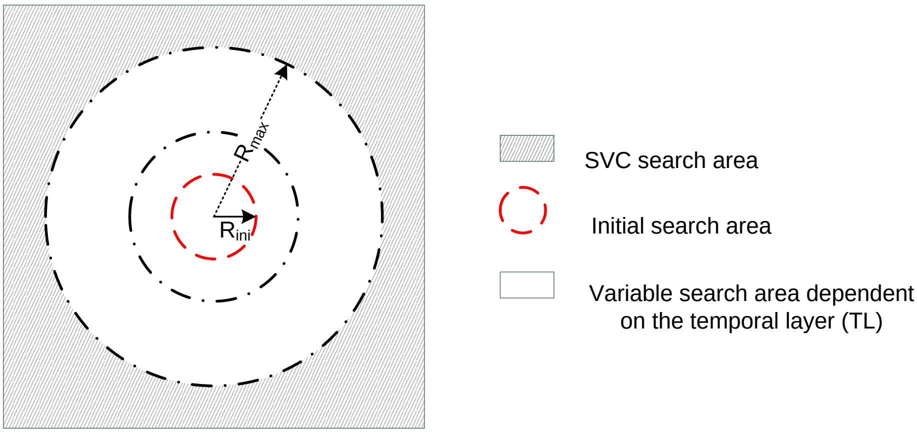


Figure 4.6. Variation of initial search area depending on temporal layer

4.3 Performance Evaluation

In this section, results from the implementation of the proposal described in the previous section are shown.

4.3.1 Time Analysis of SVC

For analyzing time consuming of SVC, JSVM reference software [104] has been used. Different sequences with varying characteristics were used with CIF and QCIF resolutions. These sequences were encoded with JSVM software with dyadic coding structure which is the used by default by this software. Different QP values were set (28, 32, 36, and 40) and the average percentage of the encoding time spent on encoding each of the temporal layers has been measured. These experiments were done with different GOP sizes for QCIF and CIF resolutions and Baseline and Main profiles. The results are shown in the following subsections.

Baseline Profile

As it is shown in Table 4.1and Table 4.2, the highest identifier has the temporal layer, the encoding time increases. Focusing on the two temporal layers with the highest identifier, approximately the 80% of the time spent on encoding the full sequence is used to encode these temporal layers. These results are represented in a graphical way in Figure 4.7.

Table 4.1. 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	13.50	27.00	54.38	1.77	6.36	13.07	26.14	52.65
City	12.96	28.89	58.16	5.12	13.49	27.02	54.37	1.78	6.38	13.07	26.15	52.62
Foreman	12.87	28.92	58.2	5.07	13.58	27.01	54.35	1.76	6.31	13.14	26.16	52.64
Soccer	12.74	29.01	58.24	5.04	13.52	27.04	54.4	1.71	6.43	13.01	26.22	52.64
Harbour	12.96	28.89	58.16	5.14	13.50	27.00	54.36	1.78	6.36	13.07	26.14	52.64
Mobile	12.88	28.91	58.21	5.09	13.53	27.02	54.37	1.75	6.39	13.09	26.14	52.62
Average	***12.89***	***28.92***	***58.19***	***5.1***	***13.52***	***27.02***	***54.37***	***1.76***	***6.37***	***13.08***	***26.16***	***52.64***

Table 4.2. Encoding time (%) for each temporal layer (TL) with different GOP sizes using CIF – Baseline Profile

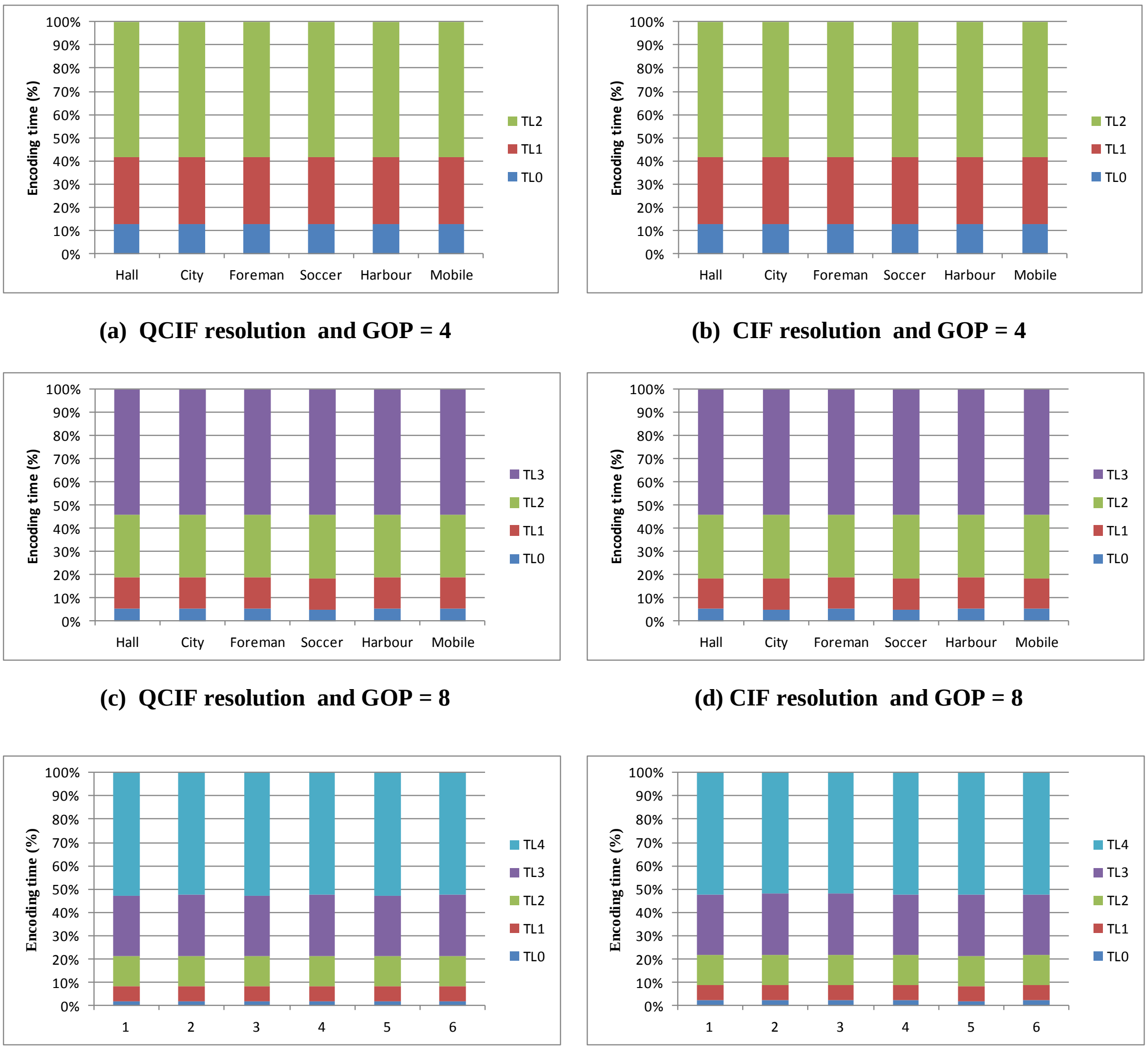
Encoding time (%) of every temporal layer – CIF (30 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	**TL0**	**TL1**	**TL2**	**TL0**	**TL1**	**TL2**	**TL3**	**TL0**	**TL1**	**TL2**	**TL3**	**TL4**
Hall	12.71	29.04	58.25	5.09	13.46	27.09	54.36	2.45	6.46	12.91	26.00	52.18
City	12.68	29.04	58.28	5.05	13.52	27.15	54.29	2.45	6.47	12.93	26.02	52.13
Foreman	12.55	29.09	58.36	5.09	13.50	27.09	54.32	2.44	6.49	12.95	25.98	52.13
Soccer	12.72	29.03	58.25	5.01	13.50	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	1.60	6.51	13.03	26.23	52.62
Mobile	12.67	29.05	58.28	5.08	13.47	27.10	54.36	2.42	6.48	12.93	26.01	52.17
Average	***12.65***	***29.06***	***58.30***	***5.07***	***13.48***	***27.11***	***54.34***	***2.29***	***6.48***	***12.96***	***26.05***	***52.23***

Main Profile

The same sequences were encoded in Main Profile to see the % of the time spent for encoding each temporal layer [100]. The results are shown in Table 4.3 and Table 4.4. It can be observed that, as in Baseline Profile, approximately the 80% of the encoding time was spent in the two temporal layers with the highest identifier. These results are represented in a graphical way in Figure 4.8.

4.3.2 Baseline Profile Scenario

For measuring the performance evaluation of the proposal, test sequences with varying characteristics were used, namely Hall, City, Foreman, Soccer, Harbour, and Mobile in CIF resolution (30 Hz) and QCIF resolution (15 Hz). These sequences were encoded using the H.264/AVC JM reference software [103], version 16.2, with an IPPP pattern with a fixed QP = 28 in a trade-off between quality and bitrate. The characteristics of the H.264/AVC bitstreams are shown in Table 4.5.



(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

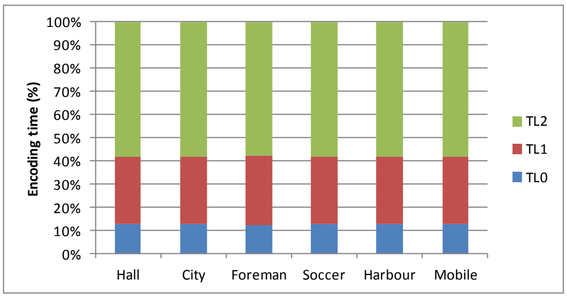
Figure 4.7. Encoding time (%) for each temporal layer with different resolutions and GOP sizes - Baseline Profile

Table 4.3. Encoding time (%) for each temporal layer (TL) with different GOP sizes using QCIF – Main Profile

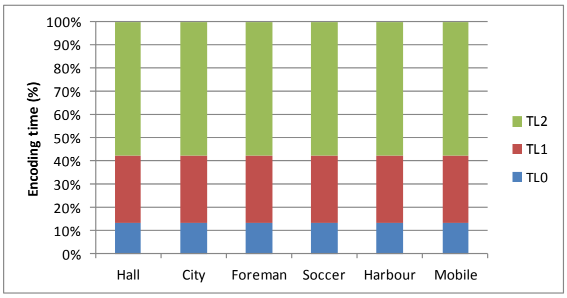
Encoding time (%) of every temporal layer – QCIF (15 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	**TL0**	**TL1**	**TL2**	**TL0**	**TL1**	**TL2**	**TL3**	**TL0**	**TL1**	**TL2**	**TL3**	**TL4**
Hall	12.86	28.92	58.23	5.08	13.52	27.1	54.3	1.77	6.36	13.07	26.14	52.65
City	12.87	28.94	58.19	5.1	13.51	27.02	54.37	1.78	6.38	13.07	26.15	52.62
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	12.70	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	27.00	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***

Table 4.4. Encoding time (%) for each temporal layer (TL) with different GOP sizes using CIF – Main Profile

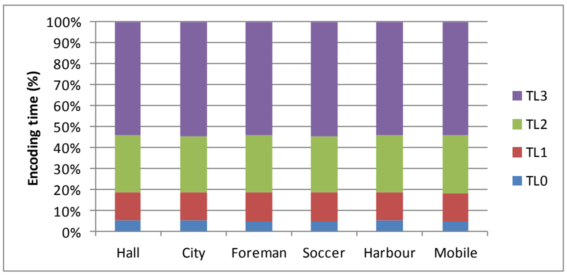
Encoding time (%) of every temporal layer – CIF (30 Hz)												
	GOP = 4			GOP = 8				GOP = 16				
Sequence	**TL0**	**TL1**	**TL2**	**TL0**	**TL1**	**TL2**	**TL3**	**TL0**	**TL1**	**TL2**	**TL3**	**TL4**
Hall	13.53	28.78	57.69	5.93	13.34	26.85	53.88	1.57	6.56	13.15	26.39	52.33
City	13.51	28.78	57.71	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	52.40
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	13.34	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	52.50
Average	***13.48***	***28.80***	***57.72***	***5.91***	***13.34***	***26.86***	***53.89***	***1.83***	***6.55***	***13.05***	***26.24***	***52.34***



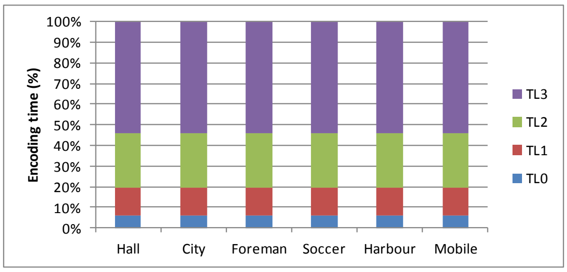
(a) QCIF resolution and GOP = 4



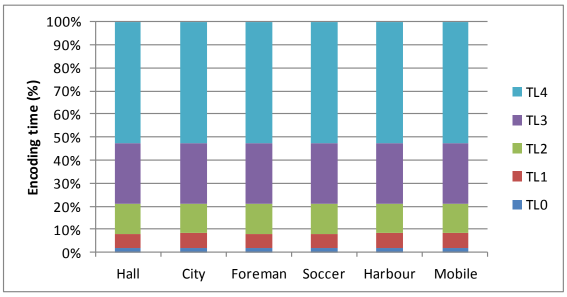
(b) CIF resolution and GOP = 4



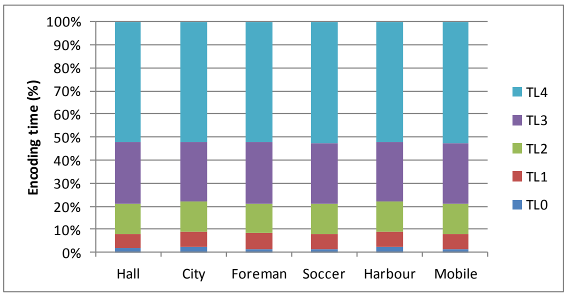
(c) QCIF resolution and GOP = 8



(d) CIF resolution and GOP = 8



(e) QCIF resolution and GOP = 16



(f) CIF resolution and GOP = 16

Figure 4.8. Encoding time (%) for each temporal layer with different resolutions and GOP sizes - Main Profile

Then, for the reference results, the encoded bitstreams are decoded and re-encoded using the JSVM software, version 9.19.3 [104] with temporal scalability, Baseline Profile and different values of QP (28, 32, 36, 40). For the results of the proposal, encoded bitstreams in H.264/AVC are transcoded using the technique described in the previous subsections and different sets of GOP length (2, 4, 8, 16, and 32) were used. The most relevant parameters used in SVC encoder configuration file are shown in Table 4.6. The remaining parameters, not shown in the table, were set to the default option.

Since the most of the SVC encoding time is spent on the temporal enhancement layers with the two highest identifiers as shown in subsection 4.3.1, our approach will be applied to these temporal layers and the remaining temporal layers will be decoded and re-encoded completely. In case there is only one temporal enhancement layer, it will be applied only to this one to avoid changes in base temporal layer. In a mathematical way, our technique will be applied to the temporal layers that satisfy the condition:

$$n = \log_2(GOP_{size}) - k,\ with\, n > 0\ and\ k \in \ 0,1 \tag{4.6}$$

where n is the identifier of the temporal layer and k varies between 0 and 1.

Table 4.5. Characteristics of the H.264/AVC video bitstreams

	Characteristics of the H.264/AVC video bitstreams		
Sequence	**Bitrate (kbit/s) QCIF**	**Bitrate (kbit/s) CIF**	**GOP**
Hall	62.99	385.92	I11P
City	98.23	605.31	I11P
Foreman	115.81	611.66	I11P
Soccer	161.75	849.83	I11P
Harbour	258.47	1827.20	I11P
Mobile	334.01	2164.46	I11P

Table 4.6. Most relevant parameters in the SVC encoder configuration file

Sequences type		
Sequence	**CIF**	**QCIF**
FrameRate	30.0	15.0
IntraPeriod	32	32
ProfileIdc	66	66
RateControlEnable	0	0
SearchMode	-1	-1
SourceWidth	352	176
SourceHeight	288	144

4.3.3 Main Profile Scenario

For evaluating the performance of the proposal regarding Main Profile, the same steps described in Section 4.3.2 were followed, although there are some changes:

- The H.264/AVC bitstream were encoded using Main Profile and IBBP pattern.
- The characteristics of the H.264/AVC bitstreams were the following depicted in Table 4.7.
- The encoding parameters of SVC are the same as in Section 4.3.2, except the ProfileIdc which was changed to 77.

Table 4.7. Characteristics of the H.264/AVC video bitstreams

	Characteristics of the H.264/AVC video bitstreams		
Sequence	**Bitrate (kbit/s) QCIF**	**Bitrate (kbit/s) CIF**	**GOP**
Hall	63.29	349.33	IBBP
City	88.91	513.22	IBBP
Foreman	116.48	547.31	IBBP
Soccer	163.85	801.92	IBBP
Harbour	223.05	1520.21	IBBP
Mobile	261.69	1796.08	IBBP

4.3.4 Metrics

The metrics used to evaluate the proposed video transcoder are the RD function (Bitrate vs. PSNR), ∆Bitrate (%), ∆PSNR (dB) and Time Saving (%). These metrics are defined in the following lines:

- *RD function:* Rate distortion gives theoretical bounds on 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 on-going 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 to show the performance results, PSNR are the distortion for a given bitrate. The averaged PSNR values of luminance (Y) and chrominance (U, V) is used in the RD function graphs. The averaged-global PSNR is based on the Equation 4.7.

$$\overline{PSNR} = \frac{4 \cdot PSNR_Y + PSNR_U + PSNR_V}{6} \qquad (4.7)$$

- *ΔPSNR (dB) and ΔBitrate (%):* The detail procedures in calculating these differences can be found from a JVT document authored by Bjøntegaard [105].

This mechanism is proposed for finding numerical averages between RD-curves as part of the presentation of results. ΔPSNR represents the difference in quality (negative means quality loss) and ΔBitrate represents the bitrate increment (positive means that bitrate increases).

- *Time Saving (%):* In order to evaluate the complexity reduction which achieves the proposal compared to the reference transcoder, the following calculation is defined to find the time differences. Let T_{ref} denote the coding time used by the H.264/AVC reference software and T_{prop} be the time taken by the algorithm proposed or the mechanism that has been evaluated; Time Saving is defined as (Equation 4.8). In T_{prop} the full computational cost for the operations needed to prepare the information for the approach is also included.
  In the proposals presented in this thesis, there are two different Time Saving calculated:
  - *Full Seq.:* It is the time reduced in the whole sequence when our proposal is applied.
  - *Partial:* It is the time reduced in the temporal layers where the proposal is applied on.

$$Time\,Saving\,(\%) = \frac{T_{ref} - T_{prop}}{T_{ref}} \cdot 100 \tag{4.8}$$

4.3.5 Results

In this section, the results for applying the proposal presented in this chapter for Baseline and Main Profile are presented.

Baseline Profile

From Table 4.8 to Table 4.12 are summarized the results for applying the proposal to different sequences in Baseline Profile with different GOP lengths (2, 4, 8, 16, and 32) and resolutions using various Q_p values between 28 and 40. As can be seen in these tables, the proposed technique is capable to reduce the computational complexity around 70% while presenting a negligible loss of video quality on average with a slight increment of bitrate.

Some resulting RD curves for the SVC bitstreams with different GOP sizes are shown in Figure 4.9 and Figure 4.10. These curves show how this technique is able to approach the RD achieved by the reference transcoder with any significant loss.

Table 4.8. 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 – 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	-0.004	0.16	61.50	91.04	0.000	0.05	57.63	86.43
City	-0.006	0.62	55.35	82.01	-0.006	0.49	49.71	74.84
Foreman	-0.009	0.37	42.92	63.71	-0.010	0.52	41.79	63.45
Soccer	-0.079	2.92	31.26	46.41	-0.069	2.94	35.82	43.27
Harbour	0.007	-0.04	60.68	89.97	0.003	-0.15	60.09	87.26
Mobile	0.003	-0.09	58.83	87.13	0.004	-0.12	56.19	84.74
Average	***-0.015***	***0.66***	***51.76***	***76.71***	***-0.013***	***0.62***	***50.21***	***73.33***

Table 4.9. 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 – 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	-0.001	0.40	75.64	87.41	0.000	0.19	77.49	89.43
City	-0.055	1.66	67.27	77.73	-0.068	3.33	63.57	73.38
Foreman	-0.006	0.81	50.72	58.76	-0.026	1.19	51.51	59.54
Soccer	-0.093	3.94	36.56	42.43	-0.126	4.92	38.88	44.99
Harbour	0.008	0.13	74.67	86.28	0.020	0.01	75.04	86.54
Mobile	-0.023	0.99	71.72	82.79	-0.018	0.89	69.77	80.45
Average	***-0.028***	***1.32***	***62.76***	***72.57***	***-0.036***	***1.76***	***62.71***	***72.39***

Table 4.10. 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 – 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	-0.011	0.59	70.51	87.46	-0.003	0.26	68.39	85.57
City	-0.075	2.06	78.50	78.50	-0.051	3.61	58.61	66.91
Foreman	0.015	0.75	45.19	56.27	-0.056	1.48	43.22	53.93
Soccer	-0.076	4.48	33.40	41.92	-0.105	4.97	36.10	42.45
Harbour	0.005	-0.18	69.61	86.29	-0.005	0.25	69.66	86.47
Mobile	-0.020	0.90	66.68	82.77	-0.018	0.98	62.73	82.90
Average	***-0.027***	***1.43***	***60.65***	***72.20***	***-0.040***	***1.93***	***56.45***	***69.71***

Table 4.11. 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 – 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	-0.012	0.51	67.48	86.62	0.066	0.42	66.51	86.09
City	-0.167	3.21	60.59	77.79	-0.181	3.28	56.98	73.81
Foreman	0.023	0.89	44.76	57.66	-0.049	1.20	44.89	58.53
Soccer	-0.107	5.02	32.65	42.30	-0.101	4.76	33.49	43.88
Harbour	0.040	0.22	66.69	85.51	0.135	-3.41	66.19	86.12
Mobile	-0.029	1.40	63.92	82.00	-0.021	1.56	61.73	80.03
Average	***-0.042***	***1.88***	***56.02***	***71.98***	***-0.025***	***1.30***	***54.97***	***71.41***

Table 4.12. RD performance and time savings of the approach for GOP = 32 and different resolutions

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	0.004	0.54	66.85	87.34	0.256	0.51	65.43	86.09
City	-0.111	2.41	60.84	79.25	-0.178	5.12	56.09	73.89
Foreman	-0.035	0.84	41.21	53.96	-0.049	1.20	44.89	58.53
Soccer	-0.136	5.87	35.11	45.72	-0.132	5.03	33.47	44.63
Harbour	0.047	0.27	66.38	86.30	0.059	0.21	65.40	86.11
Mobile	-0.095	2.74	63.60	82.95	-0.014	1.58	60.74	80.06
Average	***-0.054***	***2.11***	***55.67***	***72.59***	***-0.010***	***2.28***	***54.34***	***71.55***

Main Profile

Table 4.13 - Table 4.17 summarizes the results for applying the proposal to different sequences in Main Profile and various GOP sizes (2, 4, 8, 16, and 32) and resolutions using Q_Ps factors between 28 and 40. As in the results presented previously for Baseline Profile, the algorithm presents a reduction of computational complexity around 70% while maintaining video quality and bitrate.

Some resulting RD curves for the SVC bitstreams are shown in Figure 4.11 and Figure 4.12. In these RD curves is shown that the proposal curves are very close to the obtained from the reference transcoder.

4.3.6 Analysis

Taking into account the results presented in the previously sections, some conclusions can be extracted.

Table 4.13. 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 – Main 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	0.007	0.10	64.00	90.38	0.003	0.06	56.50	85.70
City	0.004	0.28	45.18	65.34	-0.006	0.31	42.28	64.58
Foreman	-0.008	0.37	41.83	58.62	-0.007	0.55	40.70	57.90
Soccer	-0.061	2.43	29.67	42.19	-0.055	2.79	26.55	40.76
Harbour	0.003	0.05	59.89	86.96	0.003	-0.07	55.44	83.87
Mobile	0.000	0.03	62.67	88.14	0.002	-0.05	54.98	83.43
Average	***-0.009***	***0.54***	***50.54***	***71.94***	***-0.010***	***0.60***	***46.08***	***69.37***

Table 4.14. 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 – Main 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	0.016	0.60	77.64	88.33	-0.004	0.45	77.69	89.43
City	-0.039	1.38	55.08	63.40	-0.126	3.26	59.61	67.90
Foreman	-0.028	1.08	48.07	55.26	-0.041	1.47	53.34	60.72
Soccer	-0.111	4.23	35.12	40.41	-0.135	5.64	36.81	42.65
Harbour	0.003	0.32	74.94	86.24	-0.003	0.28	74.80	86.69
Mobile	-0.022	0.85	74.95	86.20	-0.017	0.71	75.15	86.61
Average	***-0.030***	***1.41***	***60.97***	***69.97***	***-0.054***	***1.97***	***62.90***	***72.33***

Table 4.15. 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 – Main 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	0.022	0.62	72.27	89.35	0.007	0.48	68.38	85.58
City	-0.028	1.66	46.74	60.40	-0.120	3.62	49.16	61.81
Foreman	-0.010	0.92	41.83	51.83	-0.038	1.39	43.23	54.44
Soccer	-0.123	4.13	34.19	41.39	-0.111	5.74	30.60	38.76
Harbour	0.005	0.32	70.25	86.32	0.012	0.26	66.43	83.21
Mobile	-0.018	0.76	69.68	86.13	-0.018	0.77	66.21	82.84
Average	***-0.025***	***1.40***	***55.83***	***69.24***	***-0.045***	***2.04***	***54.00***	***67.77***

Table 4.16. 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 – Main 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	0.013	0.59	69.56	88.58	0.003	0.64	66.01	85.43
City	-0.108	2.73	50.16	64.01	-0.100	2.61	52.22	67.85
Foreman	-0.026	0.85	42.34	54.04	-0.035	1.30	44.99	58.56
Soccer	-0.087	4.61	31.04	39.81	-0.121	5.76	34.89	45.59
Harbour	0.023	0.33	67.03	85.45	0.011	0.31	64.40	83.43
Mobile	-0.013	0.68	67.05	85.45	-0.016	0.79	66.48	86.45
Average	***-0.033***	***1.63***	***54.53***	***69.56***	***-0.043***	***1.90***	***54.83***	***71.22***

Table 4.17. RD performance and time savings of the approach for GOP = 32 and different resolutions

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 32 – Main 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	-0.017	0.59	69.29	89.48	-0.002	0.75	64.79	85.26
City	-0.030	0.92	52.04	67.27	-0.119	5.70	47.41	62.73
Foreman	-0.010	0.93	37.39	48.45	-0.034	1.22	40.78	54.09
Soccer	-0.105	5.74	32.63	42.30	-0.142	5.96	29.09	38.87
Harbour	0.035	0.39	66.57	86.07	0.016	0.40	62.90	82.83
Mobile	-0.009	0.70	67.09	86.57	-0.015	0.82	62.61	82.45
Average	***-0.023***	***1.55***	***54.17***	***70.02***	***-0.049***	***2.48***	***51.26***	***67.71***

Both in Baseline and Main Profile, the time reduction achieved is appreciable. The time saving measured in the temporal layers where the approach is applied (partial) is around 70%. in Baseline Profile and 75% in Main Profile. The time reduction achieved in the whole sequence is around 55% in Baseline Profile and 60% in Main Profile. These complexity reductions are obtained without any significant increment of bitrate (in Baseline Profile between 0.62% in the best case and 2.28% in the worst one and in Main profile between 0.54% and 2.48%). Regarding PSNR, the presented technique reduces slightly the PSNR obtained by the reference transcoder. Despite this reduction of quality and increment of bitrate, this deviation compared to the reference transcoder is sufficient compensated by the reduction in computational complexity.

The presented technique achieves a good performance with different sequences with varying characteristics and resolutions, although can be observed that depending on the sequence the behaviour of the proposal can change.

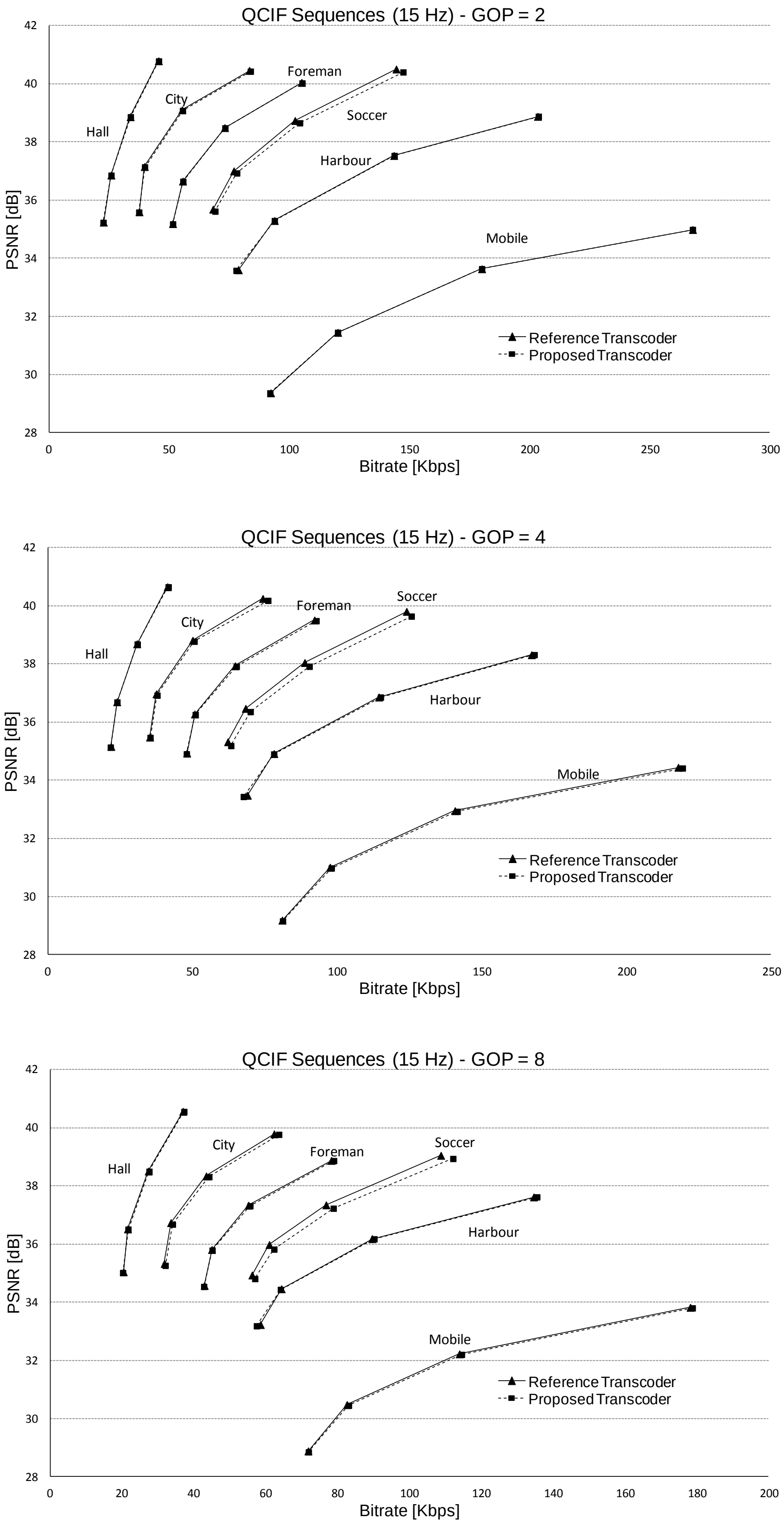


Figure 4.9. RD performance for the motion based transcoding in QCIF resolution with different GOP sizes – Baseline Profile

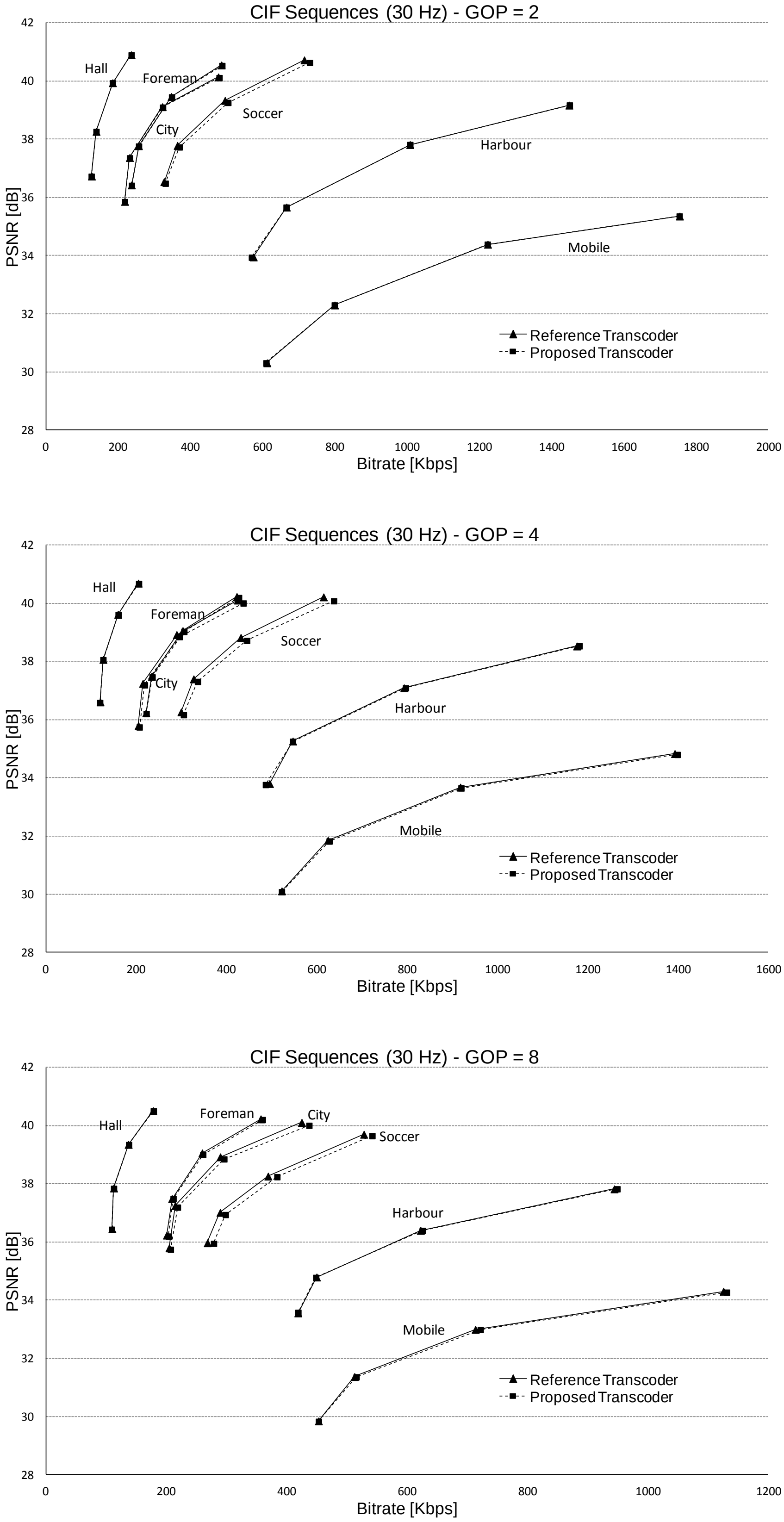
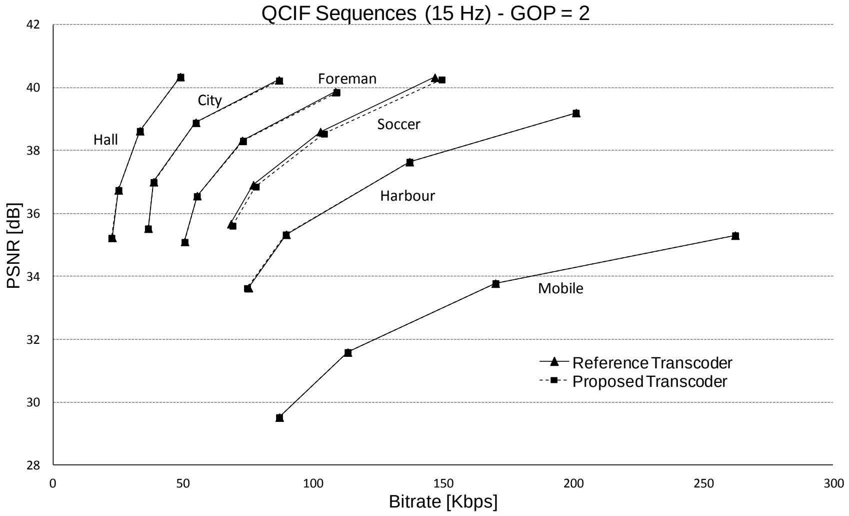
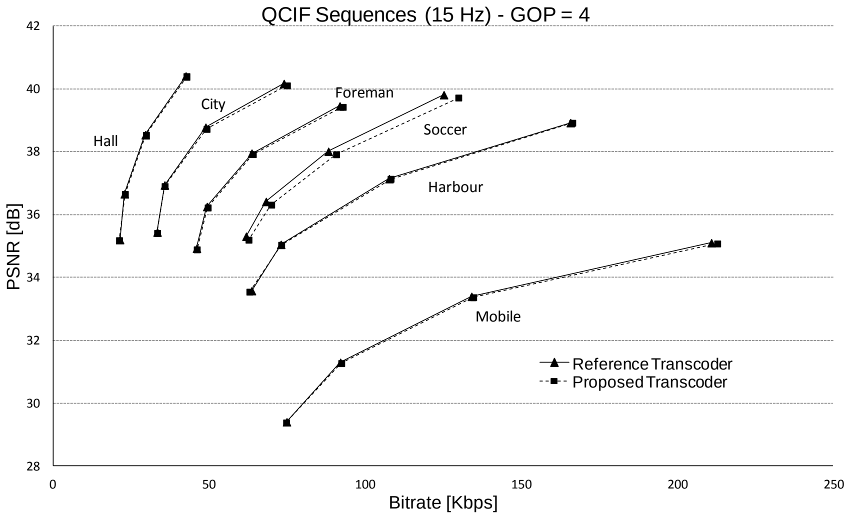


Figure 4.10. RD performance for the motion based transcoding in CIF resolution with different GOP sizes – Baseline Profile





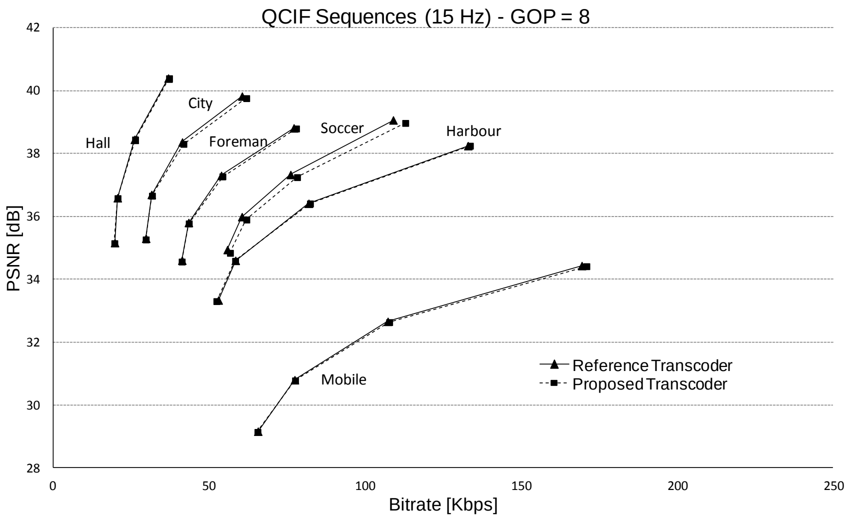
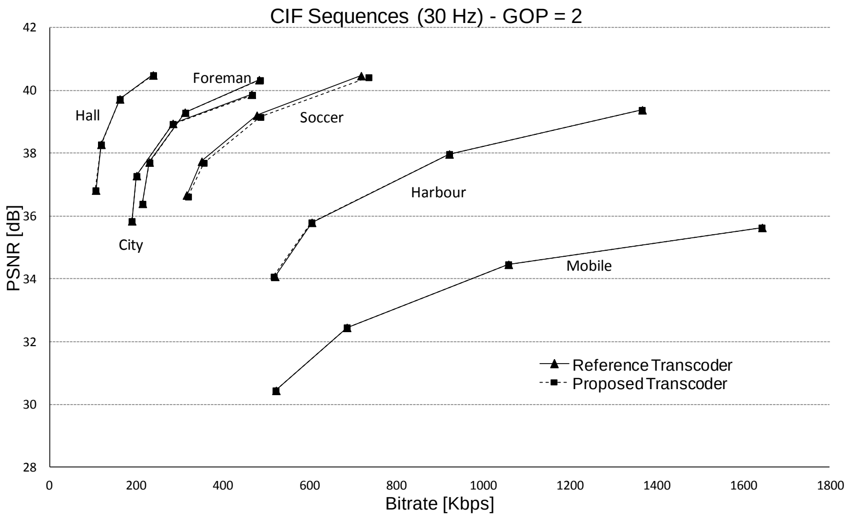
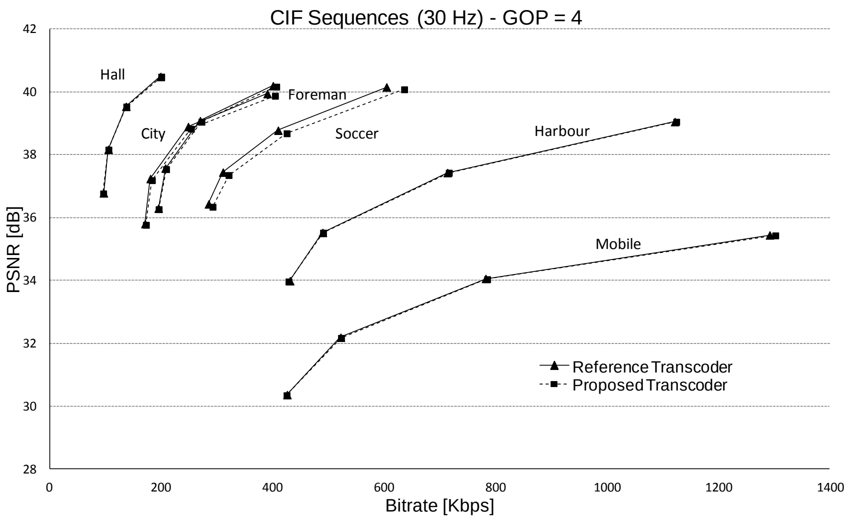


Figure 4.11. RD performance for the motion based transcoding in QCIF resolution with different GOP sizes – Main Profile





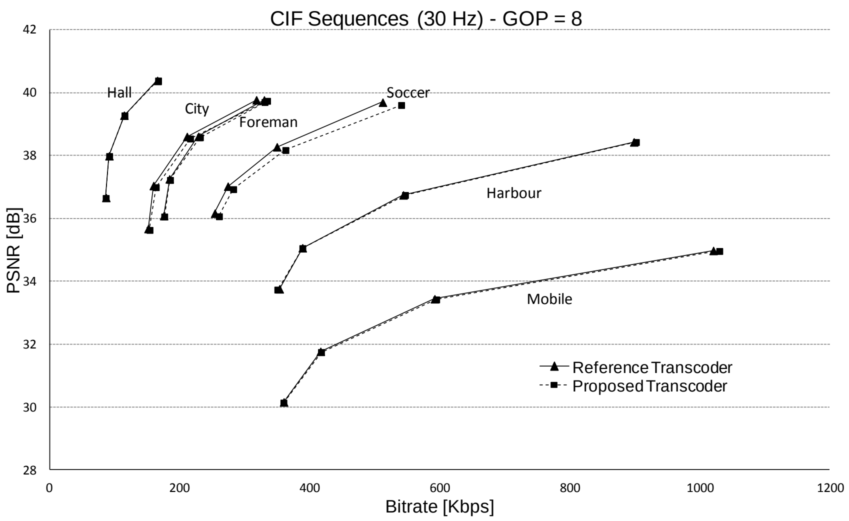


Figure 4.12. RD performance for the motion based transcoding in CIF resolution with different GOP sizes – Main Profile

For example, the time saving achieved when the proposal is applied to a sequence with high movement like *Soccer* is smaller than when the proposal is applied to sequences with soft movements like *Hall*. This is due to the length of the MVs in a sequence with high movements is larger than in a sequence with smaller movements. As the reduced search area is created using the length of these MVs, the reduction of the area is smaller if the MVs are larger, and, therefore, the time reduction compared to the reference transcoder is smaller.

Another observation is that the proposal can be applied to different GOP sizes and the results are very similar in all the cases. The impact of the GOP size and the impact of the number of the temporal layers where the proposal is applied are analyzed in following sections.

4.4 Impact of Number of Temporal Layers

In this section is presented an analysis of how many temporal layers conforms the scenario that leads to a trade-off between reduction of coding complexity and coding efficiency.

4.4.1 Scenario and Metrics

To obtain the impact of the number of temporal layers to be transcoded in our proposal, the present approach was applied on different combinations of temporal layers while the remaining layers were decoded and re-encoded completely. A fixed GOP size was chosen, GOP = 8 for QCIF resolution and 16 for CIF, so QCIF sequences were composed of four temporal layers and CIF sequences of five. This GOP selection corresponds to having a picture of the temporal base layer roughly every 0.5s. The characteristics of the sequences and the conditions of the codifications are the same as in the previous performance evaluation (see Section 4.3.2 for Baseline Profile and 4.3.3 for Main Profile).

The metrics used are Time Saving (%), ΔBitrate (%) and ΔPSNR (dB). All these metrics are defined in Section 4.3.4. In this case, only Time Saving of the full sequence is measured because the goal of this experiment is conclude how many temporal layers necessary transcoded are using the presented technique to achieve a trade-off between time saving in the whole sequence, bitrate increase and loss of PSNR

4.4.2 Results and Analysis

After decoding and re-encoding the sequences as reference and applying the technique explained previously to different combinations of temporal layers, an average of Time Saving in the whole sequence, ΔBitrate and ΔPSNR are calculated. The results for Baseline Profile are represented in Figure 4.13 and for Main Profile in Figure 4.14.

The obtained results show that by applying the proposal in different number of temporal layers, different results can be achieved, obtaining different time reductions and RD performances. For example, for QCIF resolutions, the average of ΔBitrate varies from 0.5% to near 7% depending on if the technique is applied to one temporal layer or to four. The ΔPSNR varies from near 0. dB to near 0.20 dB. The Time Saving achieved goes from 35% approximately when the proposal is applied to one temporal layer to 75% if the technique is applied to four temporal layers.

Moreover, another conclusion which can be extracted from this study is that a trade-off between time saving, bitrate increase and loss of PSNR is achieved when our approach is applied in the two temporal layers with the highest identifier as is shown in Figure 4.13 for Baseline Profile and Figure 4.14 for Main Profile.

4.5 Impact of the GOP Size

Once it has been concluded that the optimal number of temporal layers where applying the proposal is two enhancement temporal layers, another objective is determining if the technique presented is valid for different GOP sizes.

4.5.1 Scenario and Metrics

For evaluating how the GOP size influences in the technique presented, several sequences were fully decoded and re-encoded with the reference software for being used as reference and, then, the same sequences were transcoded using the algorithm described previously using different GOP sizes.

The characteristics of the sequences and the conditions of the run tests were the same as in section 4.3 for Baseline and Main Profile. The metrics used to evaluate the performance evaluation of the proposal were Time Saving (%), ΔBitrate (%), and ΔPSNR (dB). All these metrics were defined before in section 4.3.4.

4.5.2 Results and Analysis

In the results presented in Section 4.3, 4.4 and 4.5 obtained after run the codifications can be observed the different performance of the proposal with varying GOP sizes. The average of Time Saving, ΔBitrate and ΔPSNR for every GOP size are represented in a graphical way in Figure 4.15 for Baseline Profile and Figure 4.16 for Main Profile.

Both from tables of section 4.3 and from graphics can be observed that the values of ΔBitrate, ΔPSNR and Time Saving vary slightly with the GOP. This variation is due to the technique presented is applied to certain temporal layers and these layers contain different number of the total frames of the sequence depending of the GOP size. For example, in a sequence encoded with a GOP of 4, the proposal is applied to every frame which is not within the base layer, while if the sequence is encoded with a GOP length of 8, the proposal is applied to the frames within the two temporal layers with higher identifier, but there are frames in the remaining layers temporal layers (one enhancement layer and the base layer) that were decoded and re-encoded again. Although these variations exist, the minimum time reduction achieved is around 50% in the whole sequence for Baseline Profile and 45% in Main Profile, both with GOP size of 2, where the proposal is applied only to an enhancement temporal layer and the maximum value achieved of time saving is around 65% in both Baseline and Main Profile. The bitrate varies between less than 0.2% for a GOP size of 2 to 3% for a GOP size of 32. Regarding ΔPSNR, it varies from a gain of 0.2 dB to a loss of almost 0.10 dB.

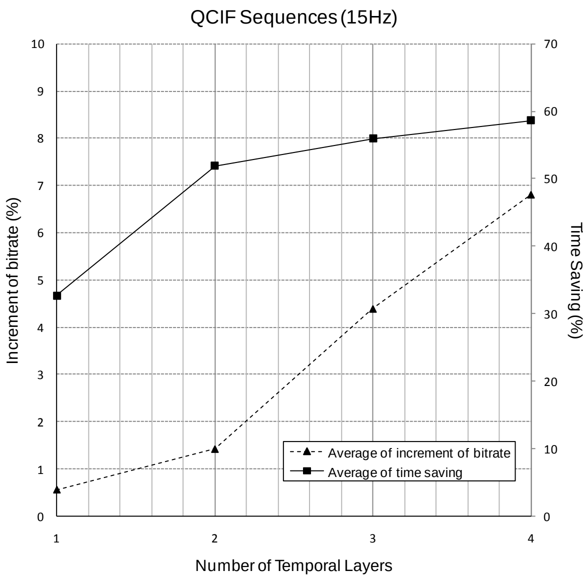
Therefore, in view of these results, it can be concluded that for both profiles the impact of the GOP size in the global results is negligible

4.6 Conclusions

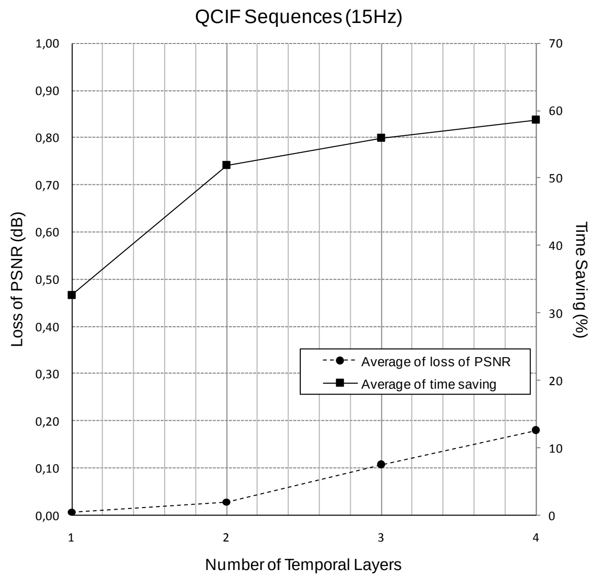
The reference transcoder decodes completely the video received and then encodes it to SVC. The most complex part of the transcoder is the encoder stage where the interprediction process takes up most of consuming resources. Focusing on the interprediction, ME is one of the tasks suitable to be accelerated.

In this chapter, as first contribution of this thesis, is presented an improved H.264/AVC-to-SVC transcoder that reduces the complexity around a 70% in the temporal layers where is applied. This improvement is achieved reusing some information collected in the decoding stage and using it for reducing the ME search area, so the encoding time is decreased.

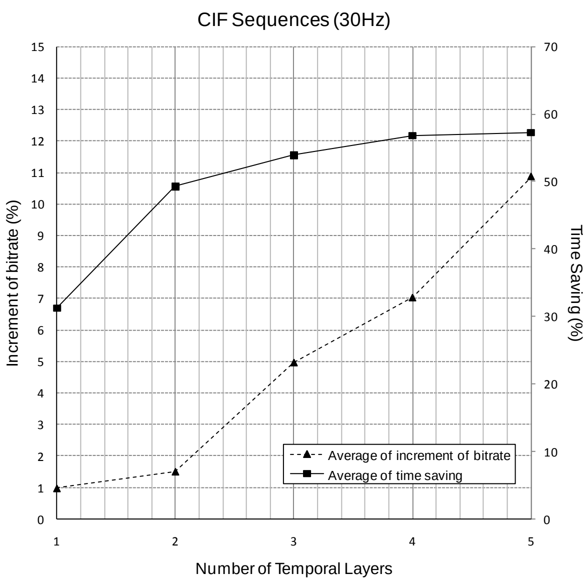
As seen from the results presented along this chapter, the algorithm presents negligible loss of video quality with a slight increment of bitrate, while experiencing an important time reduction. Moreover, it is valid for different profiles, GOP sizes and resolutions



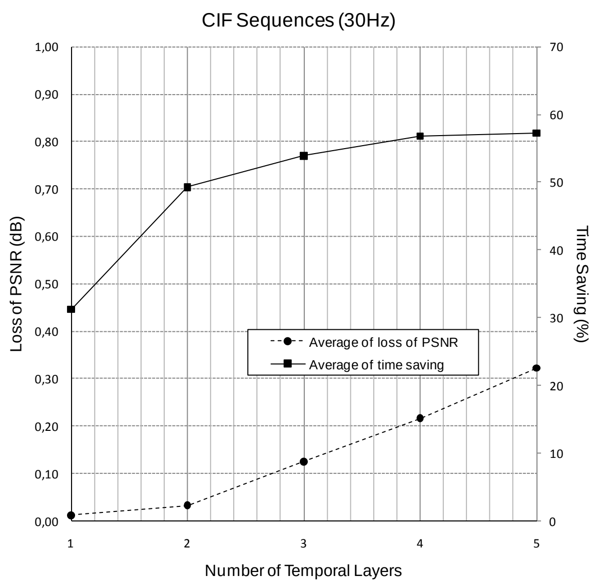
(a) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – QCIF resolution



(b) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – QCIF resolution

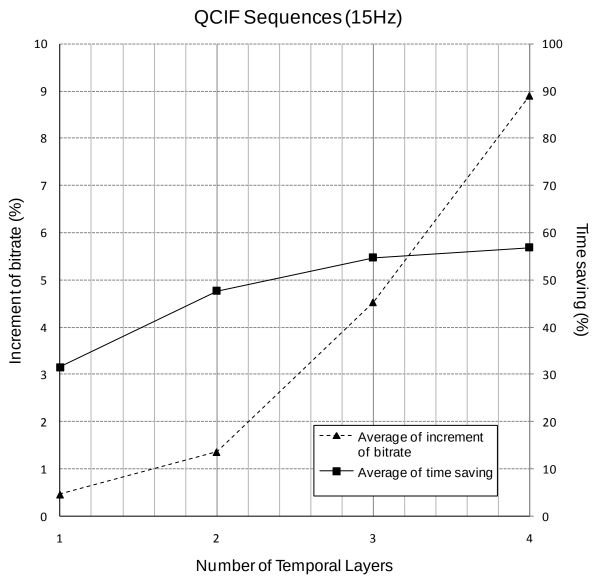


(c) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – CIF resolution

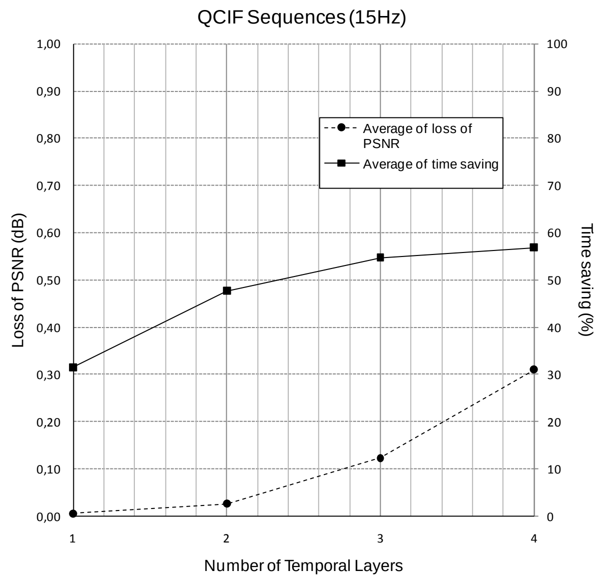


(d) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – CIF resolution

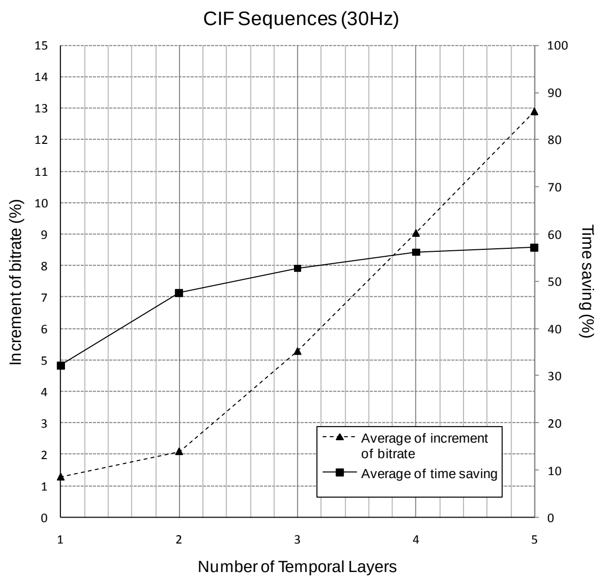
Figure 4.13. Average of increment of bitrate, loss of PSNR and time saving depending on the number of temporal layers transcoded for QCIF and CIF resolutions – Baseline Profile



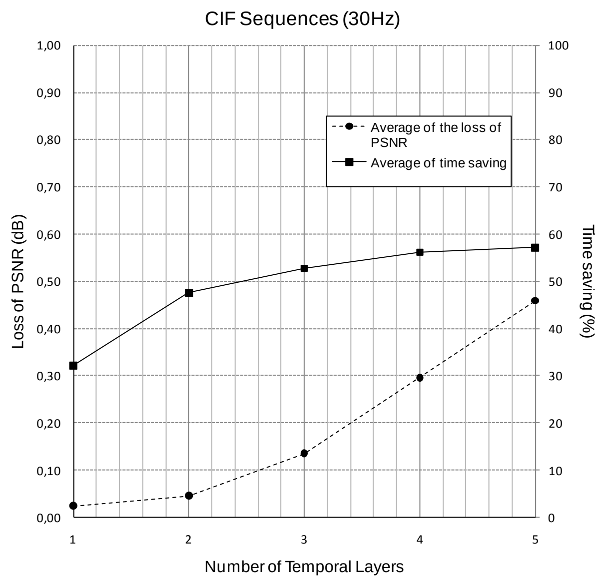
(a) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – QCIF resolution



(b) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – QCIF resolution

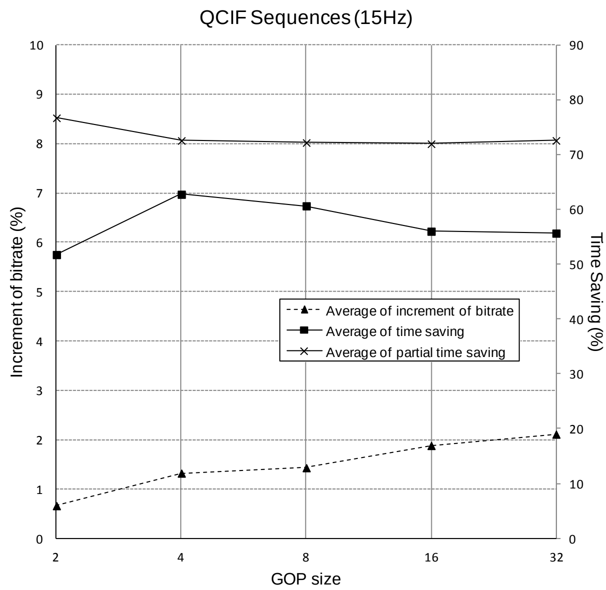


(c) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – CIF resolution

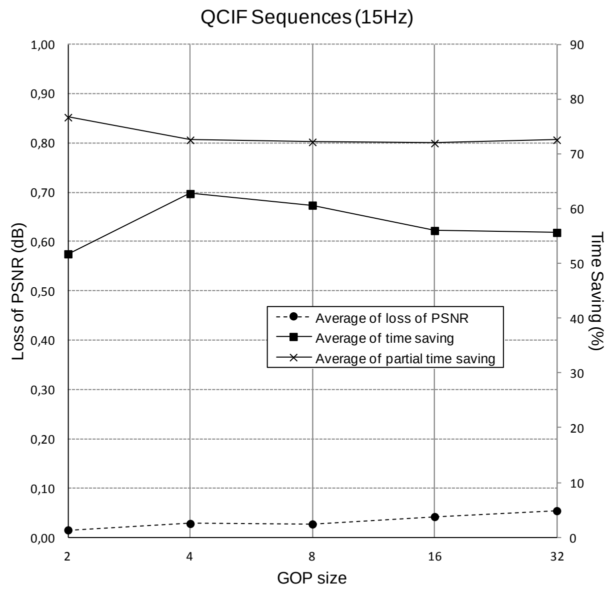


(d) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – CIF resolution

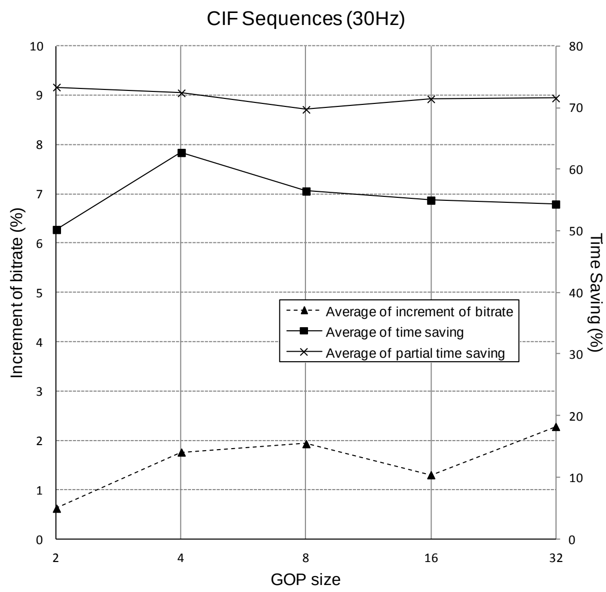
Figure 4.14. Average of increment of bitrate, loss of PSNR and time saving depending on the number of temporal layers transcoded for QCIF and CIF resolutions – Main Profile



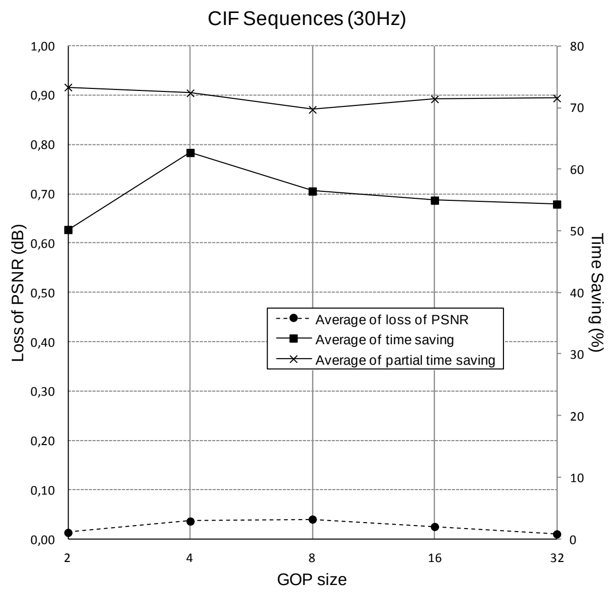
(a) ΔBitrate vs. GOP size vs. Time Saving – QCIF resolution



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

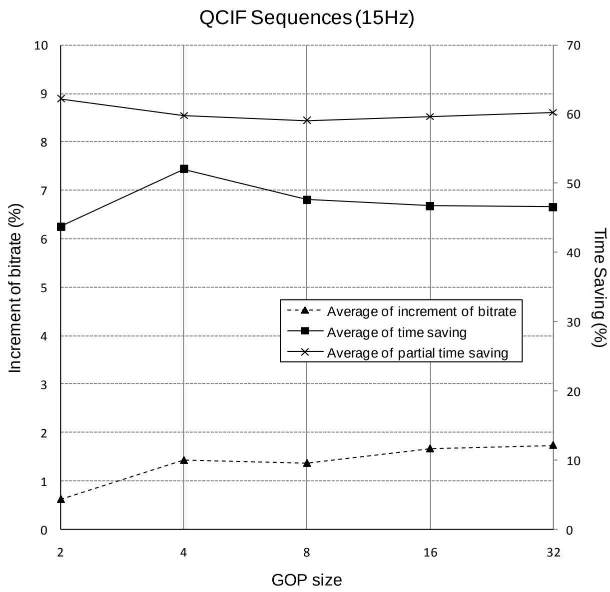


(c) ΔBitrate vs. GOP size vs. Time Saving – CIF resolution

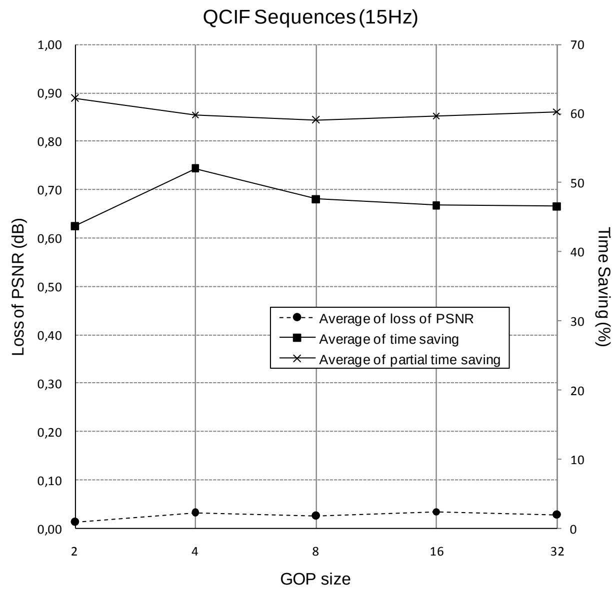


(d) Loss of PSNR vs. GOP size vs. Time Saving – CIF resolution

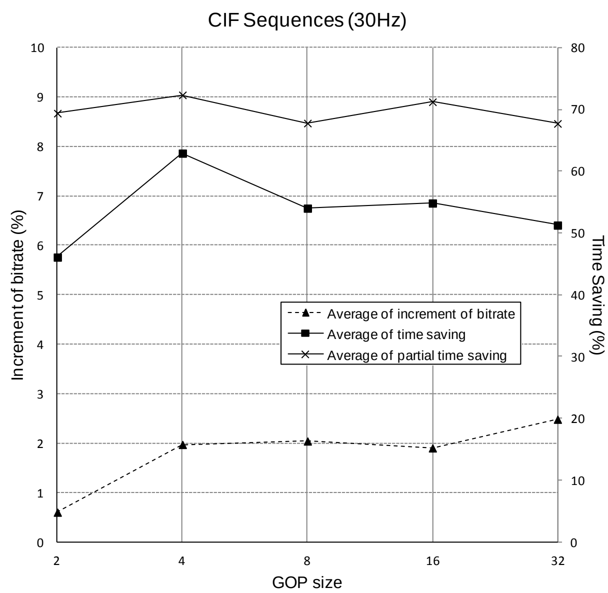
Figure 4.15. Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions – Baseline Profile



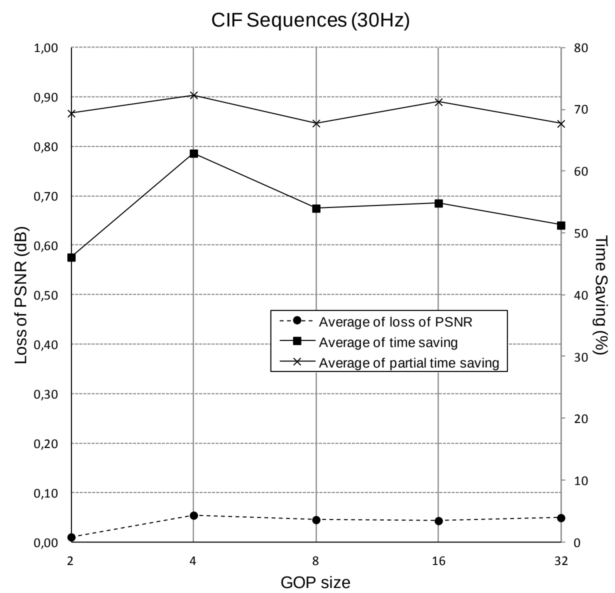
(a) ΔBitrate vs GOP size vs. Time Saving – QCIF resolution



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



(c) ΔBitrate vs. GOP size vs. Time Saving – CIF resolution



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

Figure 4.16. Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions – Main Profile

CHAPTER 5

MODE DECISION BASED H.264/AVC-TO-SVC TRANSCODING

In this chapter, the first proposal of this thesis is described. First of all, the motivation are explained, then the proposed technique is shown and, finally, a performance evaluation is done including an study of the time consuming in encoding every temporal layer and the impact of the GOP size and the number of temporal layers where the proposal is applied.

5.1 Observations and Motivation

In H.264/AVC and its extension SVC, the pictures are divided into MBs, which are further split in MB and sub-MB partitions. For every partition, a prediction is created from previously encoded data which is subtracted from the current partition 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. So in the encoding process, the encoder has to check all MB and sub-MB to determine the best option. SVC supports MC 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. This process was explained with more detail in section 2.2.2

For searching exhaustively all inter and intra modes to select the best for each MB, the SVC encoder part of the H.264/AVC-to-SVC transcoder takes a large amount of time, therefore is one of the tasks that can be accelerated for reducing the transcoding time.

Although the prediction structure (and, as a result, the frames used as a reference) of H.264/AVC without temporal scalability 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 Figure 5.1, 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, we observed that stationary areas or objects with slow motion are often coded in macroblocks without sub-blocks (such as 16x16, 16x8 or 8x16) or even as Skipped where the MB is copied from the reference one. On the other hand, the regions with sudden changes (scene, light, an object that appears) are coded in inter modes with lower MB mode partitions (such as 4x8, 8x4, 4x4) or even in 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 a more complicated MB partition such as 4x4, while shorter MVs lead to simpler MB partitions. These relationships can be observed in Figure 5.1 as well.

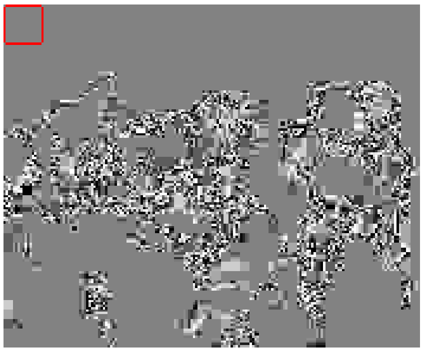
Taking into account this, it is possible to exploit this correlation using *ML* techniques [106] to build a decision tree which decides the SVC decision mode depending of the values of some information extracted from the H.264/AVC decoding stage. Thus the SVC mode decision task becomes a lookup into a decision tree with very low complexity.

For building this decision tree, the information that needs to be extracted from the H.264/AVC decoder process will be:

- Residual: The amount of residual of every block of 4x4 pixels is used by the decoder to reconstruct the decoded MB, so this information will be available in the decoding process. For our purpose, only the residual data of the luma component was extracted.
- MVs: This information is available as well in the decoding process. The MVs of each MB were extracted.
- Mode decision of H.264/AVC: The MB partitioning of each MB in H.264/AVC is related to the residual and the MVs and can give us valuable information.

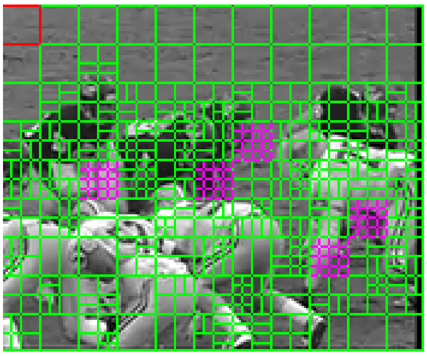
The main goal of this proposal was to reduce the time spent by this mode decision process, trying to narrow down the set of MB partitions to be checked by the encoder by using a decision tree generated by data mining techniques.

(a) Original frame

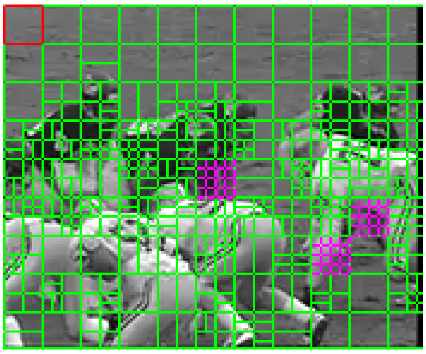


(b) Residual H.264/AVC

(c) MVs in H.264/AVC



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



(e) MB mode decision in SVC

Figure 5.1. Correlation between residual, MVs and MB mode decision

5.2 Fast MB Mode Decision Approach

In this section, a fast MB mode decision algorithm based on ML techniques is presented. This technique can be used for different profiles, resolutions and GOP sizes of SVC as it is shown in this chapter.

5.2.1 Machine Learning

ML refers to the study of algorithms and systems that learn or acquire knowledge from experiences. It uses statistics with different kind of algorithms to solve a problem by studying and analyzing the data. There are two types of learning: inductive and deductive learning. In inductive learning a synthesis of the knowledge is done, while in deductive learning an analysis of existing knowledge is performed in order to improve this knowledge and transform it into a form easier or more efficient to use. This information can be used to build a decision tree for taking decisions which is built using the training data mentioned previously. That training data must satisfy the following properties [106]:

1. Each attribute or variable can take nominal or numerical values, but the number of attributes cannot vary from a sample to another. This is to say, all the samples in the training data set used for training the model must have the same number of variables.
2. The set of categories that the samples can be assigned to must a priori be known to enable supervised learning.
3. The set of categories must be finite and must be different from one another.
4. Since the inductive learning consists of obtaining generalization from samples, it is supposed the existence of a sufficiently great number of examples.

The decision tree is made by mapping the observations about a set of data and applying a divide-and-conquer approach to the problem. It is composed by nodes represented by circles and branches which are represented by segments connecting the nodes. Routing down the tree, the end nodes are named leafs. The nodes involve testing a particular attribute. Leaf nodes give a classification that applies to all instances that reach the leaf. To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes and when a leaf is reached the instance is classified according to the class assigned to the leaf.

ML techniques has been used in an extensive range of applications including web mining, medical diagnosis, marketing and sales, speech and writing recognition, automation, identifying the genes within a new genome, etc. The use of these techniques in the areas of image and video has focused on detection of hazards or some characteristics. Moreover, in some transcoding approaches, as it was said in chapter 3, ML has been used [75][77], although these approaches focus on transcoding from several different standards to H.264/AVC.

In this thesis, ML has been used to reduce the complexity of the mode decision process in the transcoding from H.264/AVC-to-SVC proposed. In this framework, ML tools were used 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, this process can be speed up. In this thesis, a decision tree with three levels of decision is presented. This decision tree narrows down the mode decisions that can be chosen by the standard.

Figure 5.2 depicts the process for building the decision trees to be used in the H.264/AVC-to-SVC transcoding process. The H.264/AVC video is decoded and some information such as residual, MVs lengths, MB modes are saved. The decoded H.264/AVC video is then encoded using SVC standard and the coding mode of the corresponding MB is also saved. Using these data, a ML algorithm is run to create decision trees that classify an MB into one of the several SVC MB coding modes.

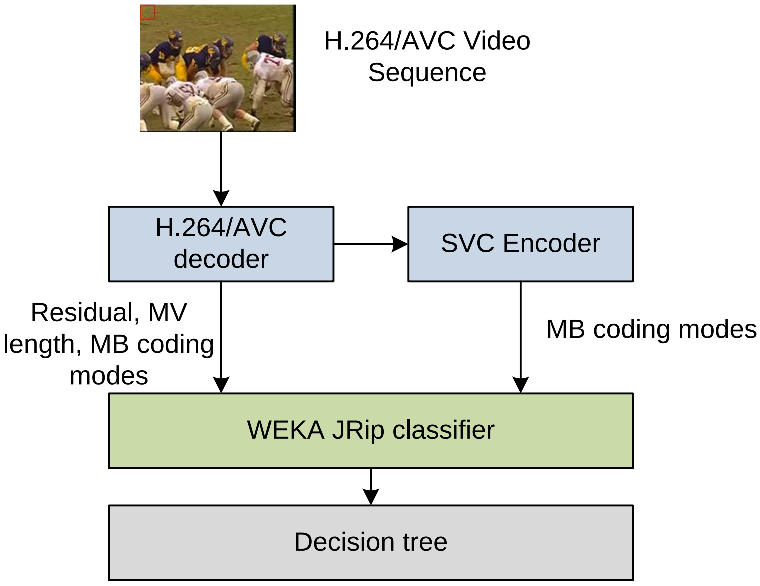


Figure 5.2. Process for building the decision tree for H.264/AVC-to-SVC transcoding

In this case, the WEKA software [107] was used. WEKA is a collection of ML algorithms for data mining tasks and also contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. A screenshot of this tool is shown in Figure 5.3.

The information gathered from H.264/AVC together with the SVC encoder mode decision was introduced in WEKA and then, an ML classifier was run. The way to introduce the datasets in WEKA is using the ARFF files. An example of ARFF file is shown in Figure 5.4. This text file contains the dataset to be classified and the relationship between a set of attributes is shown. This file has two parts:

- The header with the information about the name we give to the relation (@relation), and the definition of the attributes that are used and their types (@attribute). Nominal attributes are followed by the set of values they can take, while numeric values are followed by the keyword *numeric*.
- The data section which starts with @data which signals the starts of the instances in the dataset. The instances are written one per line, with values for each attribute, separated by commas.

Figure 5.3. Weka GUI Chooser

```
@relation 'football-2ndlevel-trainning'

@attribute vectorlengthL0 numeric
@attribute residual16x16 numeric
@attribute MBtypeAVC {0,1,2,3,8,9,10,11}
@attribute meansofvariances4x4 numeric
@attribute varianceofmeans4x4 numeric

@attribute class {0,1}

@data
16.40,1687.00,8,33.80,57.87,1
0.71,1715.00,2,95.19,68.55,1
1.00,132.00,1,1.86,0.73,0
```

Figure 5.4.. ARFF file format example

In Figure 5.4, the variable to classify is the attribute class (`@attribute class {0,1}`) which represents a set of possible MB coding modes of SVC. In this case, the decision tree developed for accelerating the mode decision will be a binary tree (this decision will be explained in the following section), so the possible values of the attribute class are '0' or '1'. The rest of attributes will be used to decide the value of the variable class and the lines below the label @data represents the values of the variables in each MB (one line for each one). The final goal is to find a simple structure to show the possible dependencies between the attribute class and the others for building a decision tree with these relationships. More details about the values of the attributes included in the ARFF files of the proposal will be provided in section 5.2.2.

This data mining procedure has to be done just once in an off-line training process. Once the knowledge has been extracted as decision tree, it will be implemented in the proposed H.264/AVC-to-SVC transcoder.

5.2.2 Approach for Baseline Profile

This section discusses the proposed fast MB mode decision algorithm for Baseline Profile [108][109].

The main idea is to build a decision tree that uses information of the decoding process of H.264/AVC and depending on these values narrow the number of MB types to be checked by the SVC encoder.

As it is said previously, using ML techniques will make possible to exploit the correlation between different variables of H.264/AVC and the MB decision mode, so in this framework, ML is used in order to convert into rules these relationships for narrowing the MB types that the SVC encoder has to check. A scheme of the proposal is shown in Figure 5.5.

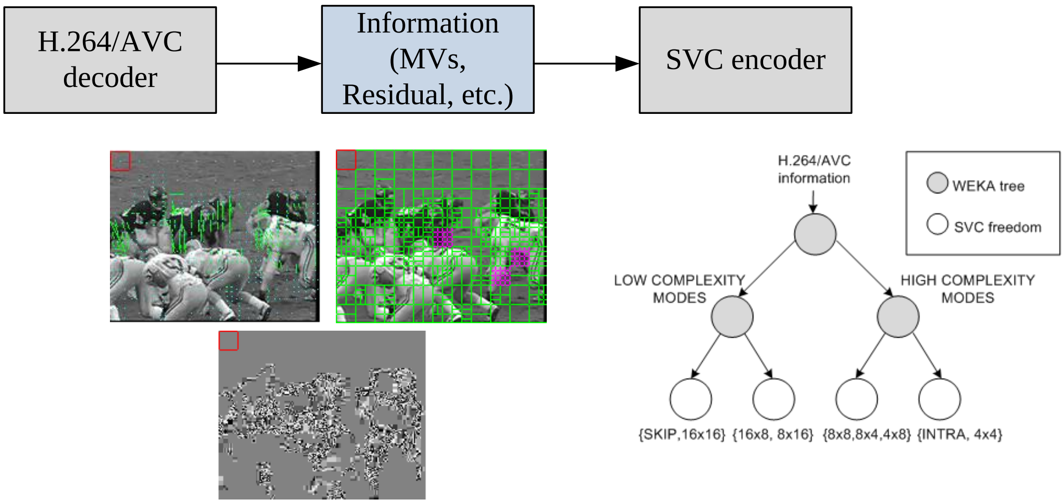


Figure 5.5. Scheme of the proposal

For every MB, the extracted information is used to generate the decision tree (and then to decide the MB partitioning). Some operations and statistics are calculated for this data. The steps for generating the decision tree are the following:

1. Extracting information per each MB in the decoder process: residual, MV length, MB type.
2. Calculating operations and statistics for these data:

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

$$residual16x16 = \sum_{i=1}^{16} res4x4_i \qquad (5.1)$$

- Length of the average of the MVs of a MB: 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 MV of the MB that we will use. Then, the length of the resulting MV is calculated.

$$MV_{xmean} = \frac{1}{n}\sum_{i=1}^{n} MV_{xi} \tag{5.2}$$

$$MV_{ymean} = \frac{1}{n}\sum_{i=1}^{n} MV_{yi} \tag{5.3}$$

$$vectorlength = \sqrt{MV_{xmean}^{2} + MV_{ymean}^{2}} \tag{5.4}$$

- Variance of means of the residual of 4x4 blocks within a MB: For every block of 4x4 pixels, the mean of the residuals of its 16 pixels (r*espixel*) is calculated (*mean4x4*). Then, the variance of these means respecting to the mean of the residual of the whole MB (*residual16x16*) is done.

$$mean4x4_i = \frac{1}{16}\sum_{j=1}^{16} respixel_j \quad \forall i \in [1,16] \tag{5.5}$$

$$varianceofmeans4x4 = \frac{1}{16}\sum_{i=1}^{16} (mean4x4_i - \overline{residual16x16})^2 \tag{5.6}$$

- Mean of variances of the residual of 4x4 blocks within a MB: For every block of 4x4 pixels, the variance of the residuals of its pixels (*respixel*) is respecting to the mean of the residuals of this 4x4 block (*mean4x4*) is calculated. Then, the mean of the variances resulting of this process is done.

$$variance4x4_i = \frac{1}{16}\sum_{j=1}^{16} (respixel_j - mean4x4_i)^2 \quad \forall i \in [1,16] \tag{5.7}$$

$$meanofvariances4x4 = \frac{1}{16}\sum_{i=1}^{16} variance4x4_i \tag{5.8}$$

3. Extracting the final MB partition of SVC as variable. As the decision tree will be a binary tree, this value will be transformed in a ‘0’ or a ‘1’ to represent in which group of each level is the MB type. The election of a binary tree is due to the possibility to exploit the similarity between groups of partitions, using the decision tree for narrowing down the partitions that the encoder has to check, but not deciding exactly the mode of the MB.

All this information was put together in an ARFF file (as in Figure 5.4) where the different variables needed where defined as attributes and each line after @data label represents the information concerning to a MB. This file is called training file and serves to generate a decision tree. For this purpose, after constructing the ARFF file with the necessary data, a classifier algorithm from the implemented in WEKA was run for obtaining the decision tree. After extensive experimentation, sequences that contain regions varying from homogeneous to high-detail serve as good training sets. In this case, *Football* QCIF sequence was used for building the training file and the classifier algorithm chosen was the JRIP algorithm [110] due to was the algorithm that obtained the best performance. Since the differences between the prediction structure of H.264/AVC without temporal scalability and the SVC prediction structure explained previously in section 4.2, the decision tree was built only using the information contained in frames within the two enhancement temporal layers with highest identifiers because the structure between the two bitstreams is very similar in these parts.

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 Figure 5.6 where the white circles represent the set of MB partition where the reference standard can choose into.

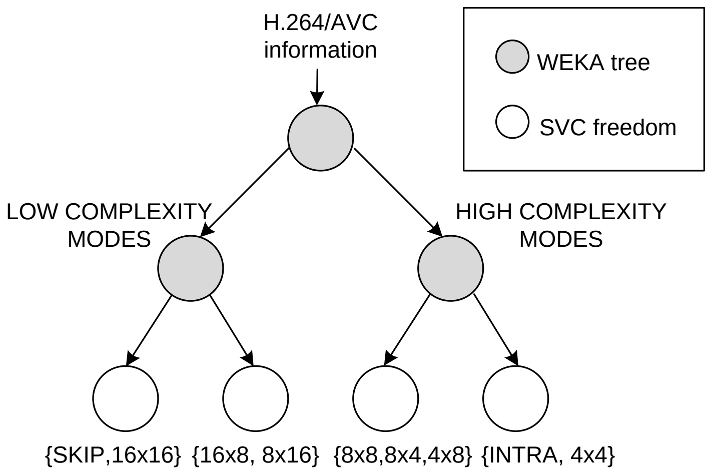
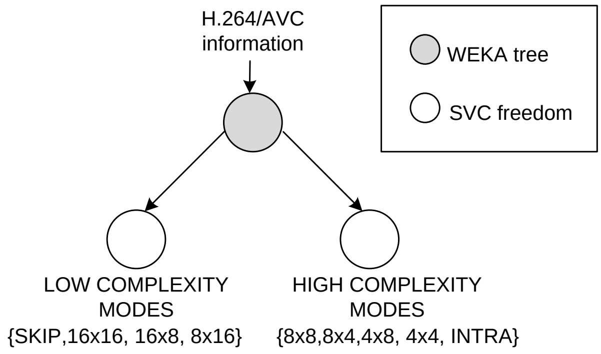


Figure 5.6. Decision tree

That final decision tree was generated for levels, taking into account the similarity between groups of partitions as was said previously. It has three levels divided as follows:

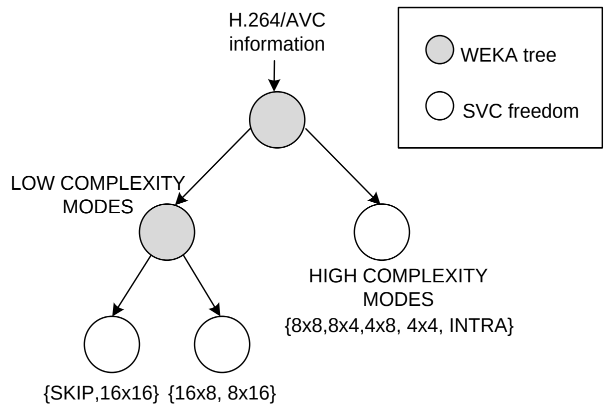
- *1st level*: Discriminates between LOW {SKIP, 16x16, 16x8, 8x16} and HIGH COMPLEXITIY {INTRA, 8x8, 8x4, 4x8, 4x4} modes.
- *2nd level*: Inside the LOW COMPLEXITY bin, a decision between {SKIP, 16x16} or {16x8, 8x16} is made.
- *3rd level*: Inside the HIGH COMPLEXITY bin, a decision between {8x8, 8x4, 4x8} or {4x4, INTRA} is made.

These different levels of the decision tree are shown in Figure 5.7, Figure 5.8, and Figure 5.9 and were implemented in the SVC encoder part of the transcoder, replacing efficiently the more complex MB coding mode decision of SVC.




```
if ((residual16x16 >= 1963) &&
   (MBtypeAVC == I4MB))
return HIGH_COMPLEXITY_MODES;

if ((varianceofmeans4x4 >= 7.71)
   && (MBtypeAVC == 8x8))
return HIGH_COMPLEXITY_MODES;

if ((varianceofmeans4x4 >= 5.16)
 && (vectorlengthL0 <= 13.89) &&
   (MBtypeAVC == 8x8))
return HIGH_COMPLEXITY_MODES;

if ((residual16x16 >= 1016) &&
   (MBtypeAVC == I4MB))
return HIGH_COMPLEXITY_MODES;

if ((residual16x16 >= 2076) &&
(varianceofmeans4x4 >= 265.07))
return HIGH_COMPLEXITY_MODES;

if ((residual16x16 >= 588) &&
   (MBtypeAVC == I4MB) &&
   (residual16x16 <= 676))
return HIGH_COMPLEXITY_MODES;

if ((varianceofmeans4x4 >= 6.78)
 && (residual16x16 >= 2278) &&
(meansofvariances4x4 <= 60.09))
return HIGH_COMPLEXITY_MODES;

if ((varianceofmeans4x4 >= 4.47)
 && (residual16x16 >= 1016) &&
(meansofvariances4x4 >= 119.55)
 && (vectorlengthL0 >= 13.45))
return HIGH_COMPLEXITY_MODES;

   In any other case:
return LOW_COMPLEXITY_MODES;
```

Figure 5.7. 1st level decision tree for Baseline Profile




```
if ((meansofvariances4x4 <= 5.4)
    && (MBtypeAVC == SKIP))
        return SKIP-16x16;

if ((varianceofmeans4x4 >=7.27)
    && (MBtypeAVC == 16x16))
        return SKIP-16x16;

if ((varianceofmeans4x4 <=
14.87) && (meansofvariances4x4
<= 6.02) && (residual16x16 <=
576) && (MBtypeAVC == 16x8))
        return SKIP-16x16;

if ((varianceofmeans4x4 <=
15.36) && (residual16x16 <= 626)
    && (MBtypeAVC == 8x16))
        return SKIP-16x16;

if ((residual16x16 <= 1551) &&
    (MBtypeAVC == 16x16) &&
  (vectorlengthL0 <= 17.69))
        return SKIP-16x16;

        In any other case:
          return 16x8-8x16;
```

Figure 5.8. 2nd level decision tree for Baseline Profile

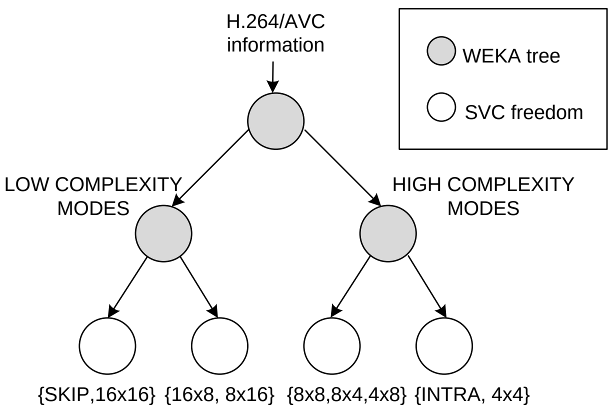
In the 1st level of the decision tree can be observed that in most of times, if the MB in H.264/AVC is partitioned in sub-MB like 8x8 or 4x4 intra, the MB type in SVC will be chosen within the HIGH COMPLEXITY modes and the same election will be done if the residual of the whole MB of H.264/AVC is large. Regarding the 2nd level of the decision tree, again the MB type collected from H.264/AVC is determinant and in the 3rd level, the variable with more influence is the length of the MVs of H.264/AVC.

This decision tree was used for mode decision task with different sequences (*Hall*, *City*, *Foreman*, *Soccer*, *Harbour* and *Mobile*) and 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 5.1.

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 QP used in the H.264/AVC stage, the residual, the mean and the variance thresholds will be different at each QP. At this point, there are two different solutions:

- Developing different decision trees for each QP and use the corresponding tree for each case.
- Developing a single decision tree and adjust the thresholds based on the QP.

The first option is rather complex because it leads to the implementation of a lot of WEKA decision trees. The solution adopted was the second one, 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 (it is shown in Figure 5.10), an adjusting in the decision tree can be done.
The proposed transcoder uses a single decision tree developed for a mid-QP of 28 which is later adjusted for other QPs (32, 36 and 40). Since the quantization step size doubles when QP increases by 6, the thresholds are adjusted by 12.5% for a change in QP of 1.




```
if ((vectorlengthL0 == 0) &&
(meansofvariances4x4 <= 76.65))
return INTRA-4x4;

if ((vectorlengthL0 == 0) &&
(varianceofmeans4x4 <= 263.27))
return INTRA-4x4;

if ((varianceofmeans4x4 >=
276.96) && (varianceofmeans4x4
<= 351.71) && (residual16x16 >=
5285))
return INTRA-4x4;

if ((varianceofmeans4x4 <=
1.63) && (vectorlengthL0 >=
37.01) && (MBtypeAVC == 16x16))
return INTRA-4x4;

if ((varianceofmeans4x4 <=
1.84) && (MBtypeAVC == 16x8))
return INTRA-4x4;

In any other case:
return 8x8-8x4-4x8;
```

Figure 5.9. 3rd level decision tree for Baseline Profile

Table 5.1. % of correct choice of MB group for baseline profile

Classification of MB groups Correct Classification (%)			
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***

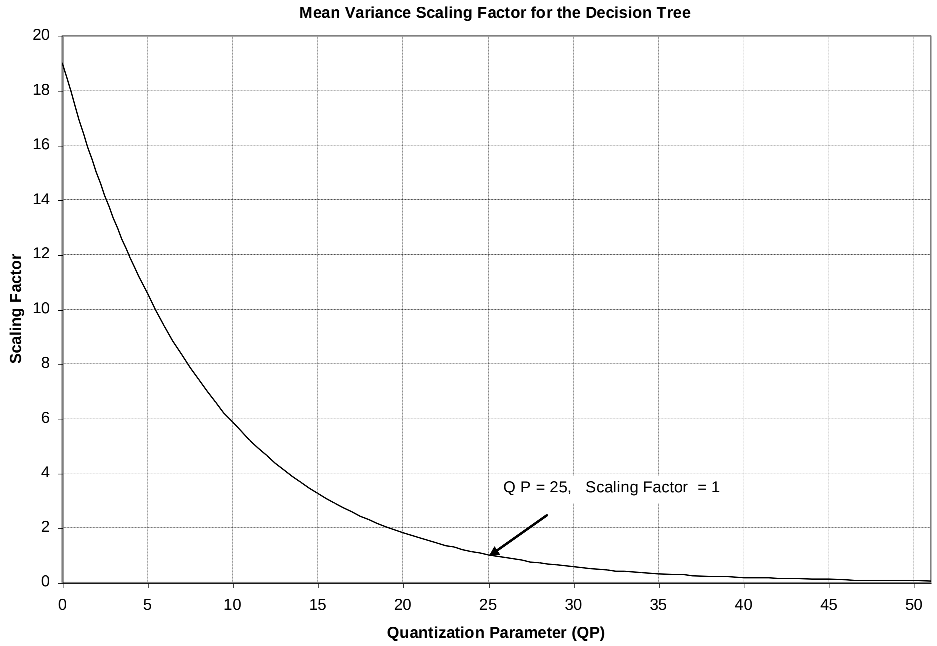


Figure 5.10. Scaling factor for the decision tree [111]

5.2.3 Approach for Main Profile

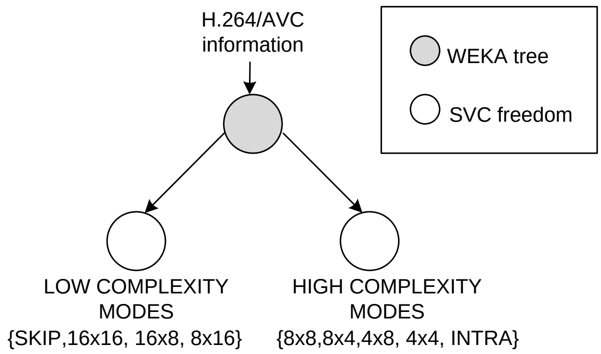
This section discusses the proposed fast MB mode decision algorithm for Main Profile. The idea is the same as in Baseline Profile (section 5.2.2), to develop a decision tree for reducing the MB modes to be checked in the SVC encoder. The process followed was the same done in Baseline Profile except that in this case the MV length of the MVs of list 1 was introduced as variable (in Baseline Profile was only used the vectors of list0).

It was necessary to develop a new decision tree for this profile because the prediction structure changes (IPPP in Baseline Profile and IBBP in Main Profile in H.264/AVC) and a new component (MVs in list1) are included. A more accurate tree for this case is built with the new conditions.

The three levels developed of the decision tree are shown in Figure 5.11, Figure 5.12 and Figure 5.13. As in the decision tree developed for Baseline Profile, in the 1st level the MB type in H.264/AVC is determinant and if the variance of means of every block of 4x4 is large, HIGH COMPLEXITY modes are selected. In the 2nd level, the variables more influential are the length of MVs of list1 and the variance of means and means of variance of every 4x4 block of the MB. In the 3rd level, if the H.264/AVC MB type is 4x4 intra, it is very possible that the MB type in SVC would be INTRA or 4x4.

The decision tree, as in Baseline Profile, was checked with different sequences (*Hall, City, Foreman, Soccer, Harbour* and *Mobile*) and classified correctly in about 91% of cases in the 1st level, 84% in the 2nd level and 90% in the 3rd level as is shown in Table

5.2. As in the case of Baseline Profile, we developed a unique decision tree that varies depending on the QP.

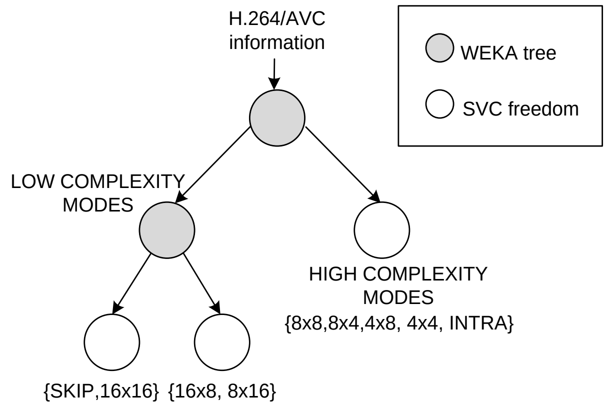



```
if ((varianceofmeans4x4 >=
27.82) && (residual16x16 >=
2373) && (MBtypeAVC == I4MB))
return HIGH_COMPLEXITY_MODES;

if ((varianceofmeans4x4 >=
15.09) && (residual16x16 >=
2006) && (MBtypeAVC == 8x8))
return HIGH_COMPLEXITY_MODES;

if (varianceofmeans4x4 >=119.18)
return HIGH_COMPLEXITY_MODES;

if ((MBtypeAVC == 8x8) &&
(varianceofmeans4x4 >= 38.02))
return HIGH_COMPLEXITY_MODES;

if ((residual16x16 >= 1591) &&
(MBtypeAVC == I4MB) &&
(meansofvariances4x4 <= 42.27)
&& (varianceofmeans4x4 >=
39.47))
return HIGH_COMPLEXITY_MODES;

if ((meansofvariances4x4 >=
9.39) && (residual16x16 >= 1063)
&& (vectorlengthL1 >= 4713.57)
&& (MBtypeAVC == I4MB))
return HIGH_COMPLEXITY_MODES;

In any other case:
return LOW_COMPLEXITY_MODES;
```

Figure 5.11. 1st level decision tree for Main Profile

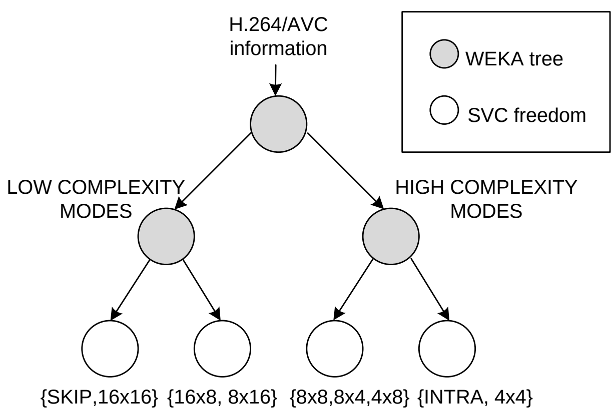
Table 5.2. % of correct choice of MB group for main profile

Classification of MB groups Correct Classification (%)			
Sequence	**1st level**	**2nd level**	**3rd level**
Hall	96.10	96.56	95.94
City	96.49	86.87	96.36
Foreman	92.26	81.83	82.14
Soccer	86.52	83.15	74.64
Harbour	88.75	77.33	91.24
Mobile	86.00	78.96	99.11
Average	***91.02***	***84.12***	***89.91***




```
if ((varianceofmeans4x4 >= 9.09)
  && (residual16x16 >= 1656))
        return 16x8-8x16;

  if ((meansofvariances4x4 >=
6.77) && (meansofvariances4x4 <=
30.44) && (varianceofmeans4x4 >=
             21.6))
        return 16x8-8x16;

  if ((meansofvariances4x4 >=
 4.02) && (varianceofmeans4x4 >=
  7.34) && (vectorlengthL1 >=
  4.14) && (vectorlengthL1 <=
            75.95))
        return 16x8-8x16;

  if ((meansofvariances4x4 >=
 4.71) && (MBtypeAVC == 8x8) &&
 (meansofvariances4x4 >= 12.58)
  && (vectorlengthL1 <= 53) &&
  (vectorlengthL0 <= 142.45))
        return 16x8-8x16;

       In any other case:
       return SKIP-16x16;
```

Figure 5.12. 2nd level decision tree for Main Profile




```
   if ((MBtypeAVC == I4MB) &&
 (meansofvariances4x4 <= 76.18)
 && (vectorlengthL0 >= 4713.57))
        return INTRA-4x4;

   if ((MBtypeAVC == I4MB) &&
 (varianceofmeans4x4 <= 183.44))
        return INTRA-4x4;

   if ((meansofvariances4x4 <=
   4.28) && (residual16x16 >=
             256))
        return INTRA-4x4;

       In any other case:
       return 8x8-8x4-4x8;
```

Figure 5.13. 3rd level decision tree for Main Profile

5.3. Performance Evaluation

In order to evaluate the fast MB mode decision approach described previously, the proposal has been implemented in a SVC encoder based on JSVM software. The results of this implementation are shown in this section.

5.3.1 Scenario and Metrics

Experiments were conducted to evaluate the performance of the proposed algorithm when transcoding videos using test sequences with varying characteristics. The characteristics of the sequences and the conditions of the experiments are the same as in the previous performance evaluations (depicted in Section 4.3.2 for Baseline Profile and 4.3.3 for Main Profile). Moreover, as in Chapter 4, the proposal has been applied on the two enhancement temporal layers with highest identifier.

The metrics used to evaluate the performance of the proposal are the RD function, Time Saving (%), ΔBitrate (%) and ΔPSNR (dB). All these metrics have been defined previously in Section 4.3.4.

This performance evaluation includes a new metric which allows comparing visually the MB mode decision chosen by the decision tree and the MB decision generated by the SVC encoder. A grid image showing the MB modes overlaid on a corresponding frame is used to this comparison.

5.3.2 Results

In this section, the results for applying the proposal presented in this chapter for Baseline and Main Profile are presented.

Baseline Profile

Table 5.3 - Table 5.7 summarizes the results (Time saving, ΔPSNR, and ΔBitrate) for applying the proposal to the different sequences in Baseline Profile with different GOP sizes (2, 4, 8, 16 and 32) and resolutions using Q_P factors between 28 and 40 according to [105]. As can be seen in these tables, the algorithm presents negligible loss of video quality on average with slight increment in bitrate. This negligible drop in rate-distortion performance is sufficiently compensated by the reduction in computational complexity (around 84%).

Figure 5.14 and Figure 5.15 show some resulting RD curves for the SVC bitstreams with several GOP sizes. In this curves it can be seen that the presented proposal for transcoding is able to approach the RD-optimal transcoded (re-encoded) reference without any significant loss.

Table 5.3. 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 - 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	0.042	-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***

Table 5.4. 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- 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	0.219	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 5.5. 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- 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	0.158	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***

Table 5.6. 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- 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	0.325	0.58	69.47	85.97	-0.673	1.90	67.61	86.89
City	-0.040	3.14	69.01	85.45	-0.140	1.96	67.16	86.39
Foreman	-0.333	3.36	65.30	82.47	-0.104	4.86	66.74	85.88
Soccer	0.068	3.03	66.02	81.60	0.031	3.60	65.29	83.83
Harbour	0.199	0.99	65.13	80.51	0.280	2.43	62.78	81.15
Mobile	0.024	1.18	63.31	79.07	0.218	0.17	63.41	81.66
Average	***0.041***	***2.05***	***66.37***	***82.51***	***-0.065***	***2.49***	***65.50***	***84.30***

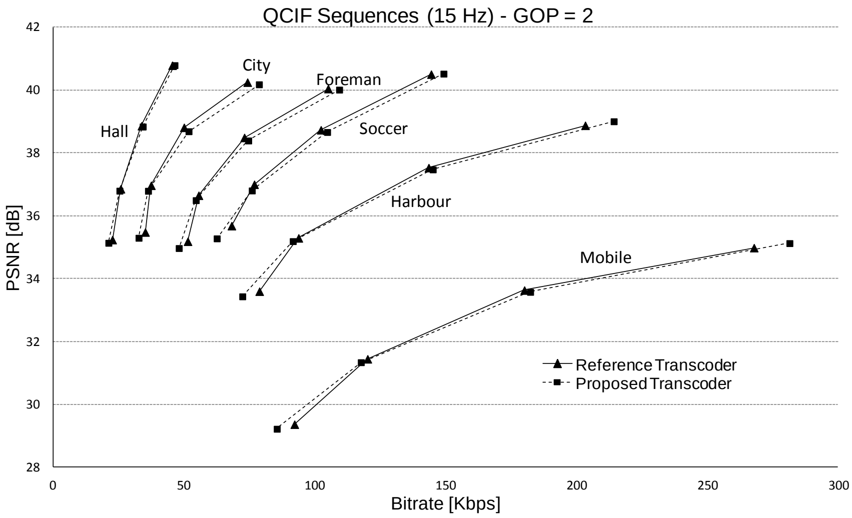
Table 5.7. RD performance and time savings of the approach for GOP = 32 and different resolutions

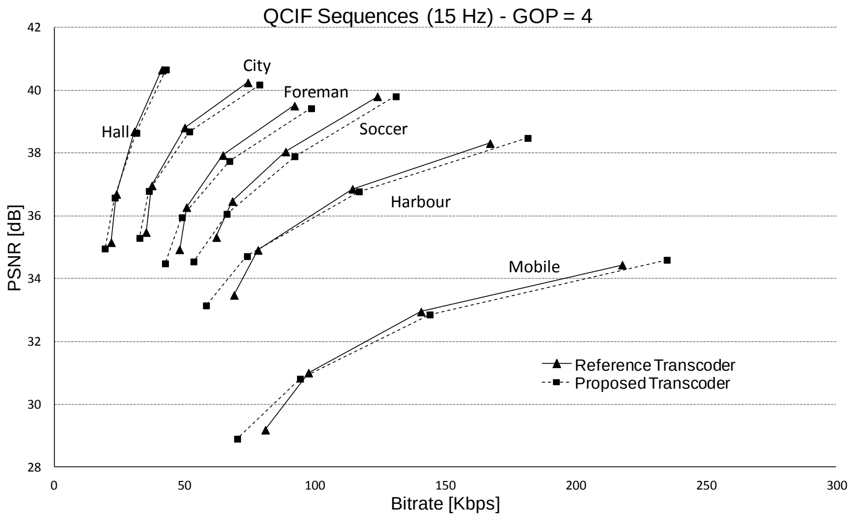
RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 32- Baseline Profile								
	QCIF (15 Hz)				***CIF (30 Hz)***			
Sequence	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
			Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	0.291	1.16	66.84	86.94	0.756	1.17	66.62	86.39
City	-0.192	3.64	66.89	85.93	-0.104	2.77	66.27	85.99
Foreman	-0.116	5.51	63.75	82.56	-0.264	5.31	66.05	85.59
Soccer	0.073	4.53	63.92	81.96	0.019	3.97	64.59	83.52
Harbour	0.122	2.41	61.86	80.05	-0.009	2.46	62.64	81.25
Mobile	0.039	2.25	61.59	79.70	0.158	1.62	62.65	81.46
Average	***0.036***	***3.25***	***64.14***	***82.86***	***0.093***	***2.88***	***64.80***	***84.03***

Figure 5.14 and Figure 5.15 show some resulting RD curves for the SVC bitstreams with several GOP sizes. In this curves it can be seen that the presented proposal for transcoding is able to approach the RD-optimal transcoded (re-encoded) reference without any significant loss. Finally, Figure 5.16 shows the difference between the MB partitioning made by the reference transcoder and the proposed algorithm, with a Q_P value of 28 in sequences *Foreman* and *City*. Both encoding processes were run under the same conditions. It can be observed that the partitioning is not exactly the same, but they are very similar and the penalty in bitrate and PSNR is minimal maintaining the coding efficiency, but reducing significantly the time needed.

Main Profile

Table 5.8 - Table 5.12 summarizes the results (Time saving, ΔPSNR, and ΔBitrate) for applying the proposal to the different sequences in Baseline Profile with different GOP sizes (2, 4, 8, 16 and 32) and resolutions using Q_P factors between 28 and 40.





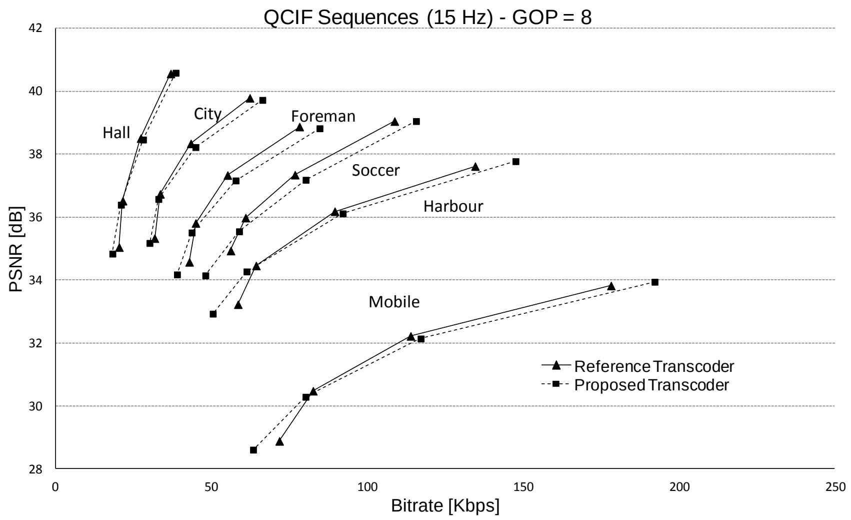
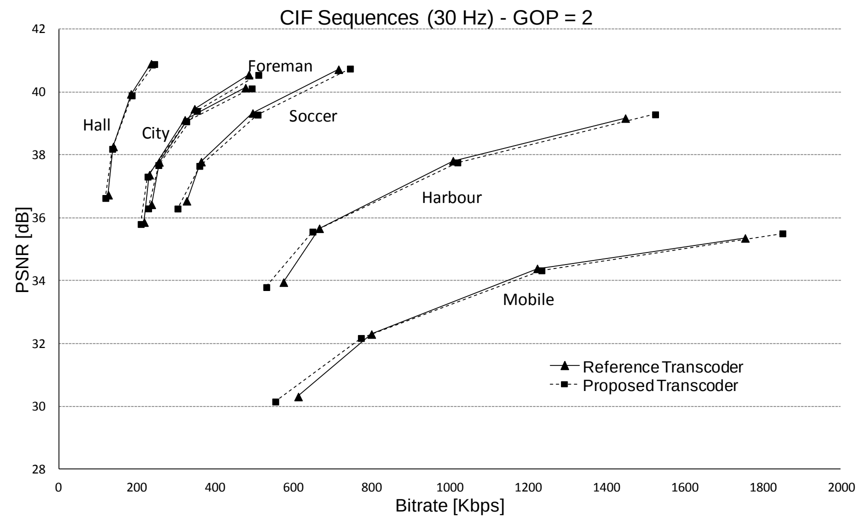
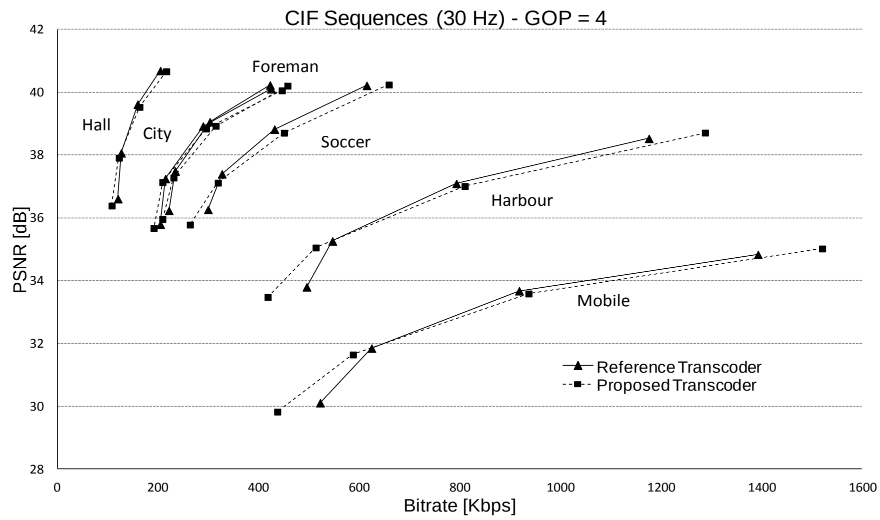


Figure 5.14. RD performance for the motion based transcoding in QCIF resolution with different GOP sizes – Baseline Profile





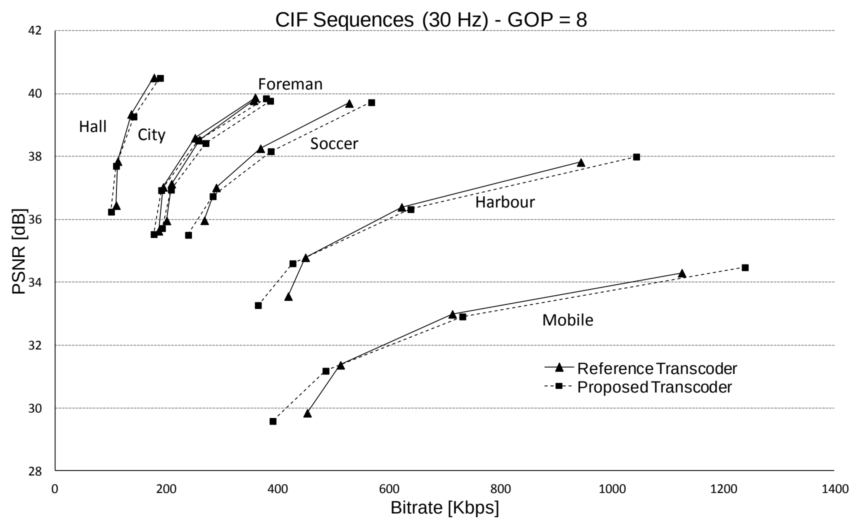
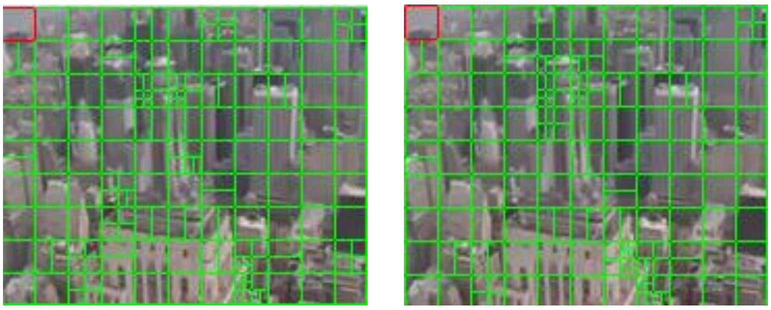
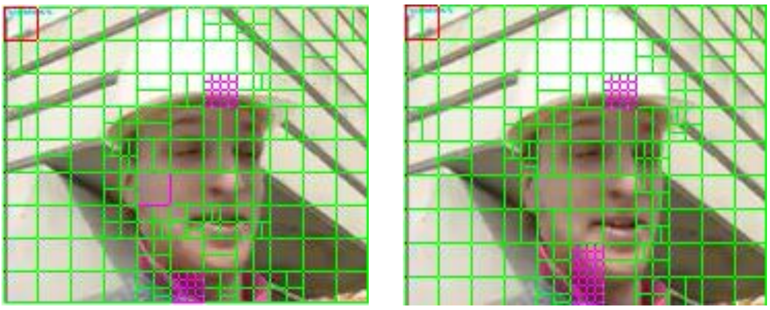


Figure 5.15. RD performance for the motion based transcoding in CIF resolution with different GOP sizes – Baseline Profile



(a) 2nd frame of City sequence (1st P-frame)



(b) 2nd frame of Foreman sequence (1st P-frame)

Figure 5.16. MB partitioning for the proposed H.264-to-SVC transcoder in Baseline Profile (left) compared to the reference one (right).

As in the previous results for Baseline Profile, it can be seen in these tables, the algorithm presents negligible loss of video quality on average with slight increment in bitrate. This negligible drop in rate-distortion performance is sufficiently compensated by the reduction in computational complexity (around 82%).

Some resulting RD curves for the SVC bitstreams with several GOP sizes are shown in Figure 5.17 and Figure 5.18 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 very close to the results obtained when applying the reference transcoder (re-encoder) while a significant reduction of computational complexity is achieved (around an 85% where the proposal is applied).

As in Baseline Profile, the difference between the MB partitioning made by the reference transcoder and the proposed algorithm, with a QP value of 28 in sequences *Foreman* and *City* is shown in Figure 5.19. It can be observed, as well as in Baseline Profile, that in both cases the MB portioning is very similar.

Table 5.8. 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- Main 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	0.064	-0.39	58.13	87.06	0.069	-0.48	58.00	85.84
City	0.061	0.34	59.70	87.92	0.049	-0.41	59.39	88.15
Foreman	-0.004	1.51	57.17	86.76	-0.039	1.42	59.74	88.30
Soccer	-0.016	3.46	62.28	81.24	0.133	1.16	59.36	88.10
Harbour	0.068	-0.52	58.18	85.36	0.055	-0.77	57.20	84.90
Mobile	0.016	-0.13	57.83	85.05	-0.005	1.71	58.89	86.90
Average	***0.032***	***0.71***	***58.88***	***85.57***	***0.044***	***0.44***	***58.76***	***87.03***

Table 5.9. 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- Main 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	0.166	0.15	77.92	88.55	0.230	-1.30	76.24	88.17
City	0.141	2.07	77.47	88.49	0.017	0.36	76.29	88.23
Foreman	0.048	3.46	77.21	88.08	-0.016	3.20	76.51	88.35
Soccer	-0.095	5.42	74.86	85.97	0.079	3.45	75.47	87.06
Harbour	0.171	-0.40	75.56	86.49	0.136	-0.83	74.56	86.35
Mobile	0.041	0.76	74.49	85.66	0.097	-0.63	74.20	85.94
Average	***0.079***	***1.91***	***76.25***	***87.21***	***0.091***	***0.71***	***75.55***	***87.35***

Table 5.10. 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- Main 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	0.668	-0.07	71.66	87.93	0.443	-1.09	71.20	88.23
City	0.063	1.81	71.12	87.99	0.018	0.31	71.29	88.31
Foreman	0.040	3.39	72.65	88.41	-0.171	3.21	71.47	88.42
Soccer	-0.027	5.52	69.99	86.09	0.105	3.46	70.57	87.24
Harbour	0.361	-0.46	70.30	86.57	0.244	-0.73	69.98	86.48
Mobile	0.022	0.68	70.39	86.31	0.212	-0.23	71.63	87.28
Average	***0.188***	***1.81***	***71.02***	***87.22***	***0.142***	***0.82***	***71.02***	***87.66***

Table 5.11. 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- Main 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	0.632	-0.06	68.34	86.72	0.337	-0.89	66.81	87.92
City	-0.009	1.92	68.91	87.18	-0.008	0.34	68.75	88.37
Foreman	0.096	2.72	67.62	86.39	-0.102	2.99	68.93	88.46
Soccer	-0.029	5.06	67.15	85.06	0.117	3.66	68.13	87.36
Harbour	0.172	-0.42	66.72	85.27	0.168	0.54	67.80	87.09
Mobile	0.033	0.53	67.29	85.28	0.185	0.10	68.39	87.22
Average	***0.149***	***1.63***	***67.67***	***85.98***	***0.116***	***1.12***	***68.14***	***87.74***

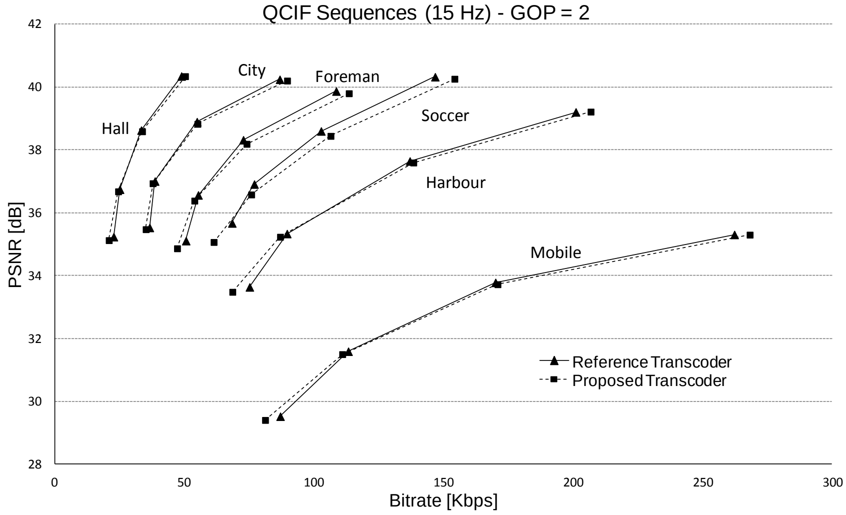
Table 5.12. RD performance and time savings of the approach for GOP = 32 and different resolutions

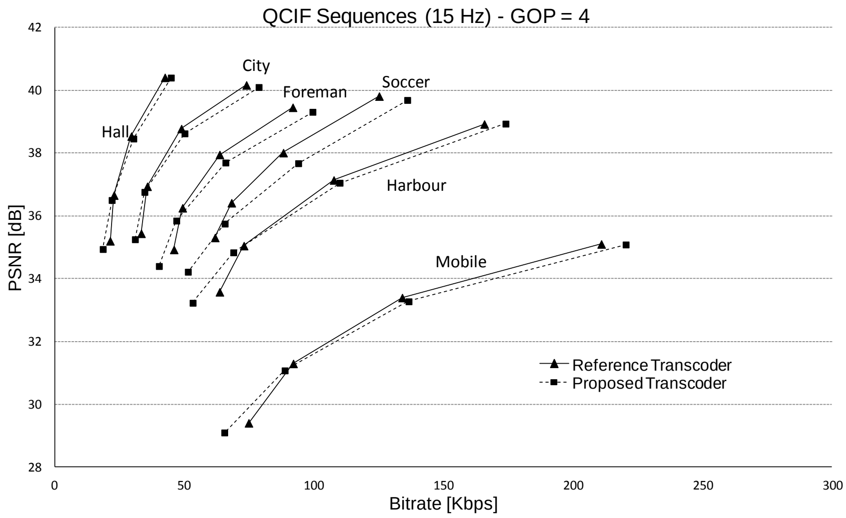
RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 32- Main Profile								
	QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
			Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	0.626	0.29	70.31	88.83	0.166	-0.34	66.90	87.78
City	-0.020	1.68	68.97	88.61	-0.011	0.46	67.60	87.78
Foreman	-0.024	3.41	69.16	88.32	0.001	3.19	67.93	87.93
Soccer	0.018	5.79	66.12	86.12	0.089	3.62	67.17	86.92
Harbour	0.371	0.16	68.13	87.39	0.229	-0.15	66.96	86.79
Mobile	0.102	1.03	67.08	86.74	0.144	0.46	66.63	86.59
Average	***0.179***	***2.06***	***68.30***	***87.67***	***0.103***	***1.21***	***67.20***	***87.30***

5.3.3 Analysis

Analyzing the results shown in the previous subsection, some conclusions can be extracted.

Both in Baseline and Main Profile the reduction of computational complexity is appreciable. The Partial Time Saving (the time reduction measured only in the temporal layers where the proposal is applied) achieved is around 84% for Baseline and 87% for Main Profile. Regarding Total Time Saving (the time reduction measured in the whole sequence) a reduction of 65% for Baseline and a 68% for Main profile are achieved. These time reductions are obtained without any significant increment of bitrate (in Baseline Profile between 0.28% in the best case and 3.25% in the worst one and in Main Profile between 0.44% and 2% in the worst case). About PSNR, the presented algorithm improves the PSNR obtained by the reference transcoder. This is possible because, both reference and proposed transcoder are encoded with the RDO disable, so the encoding made by the reference is not the most optimized one.





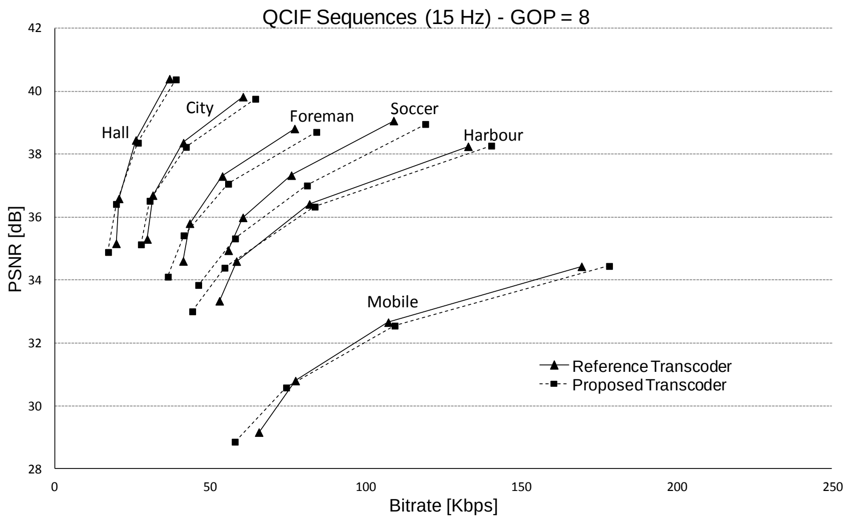
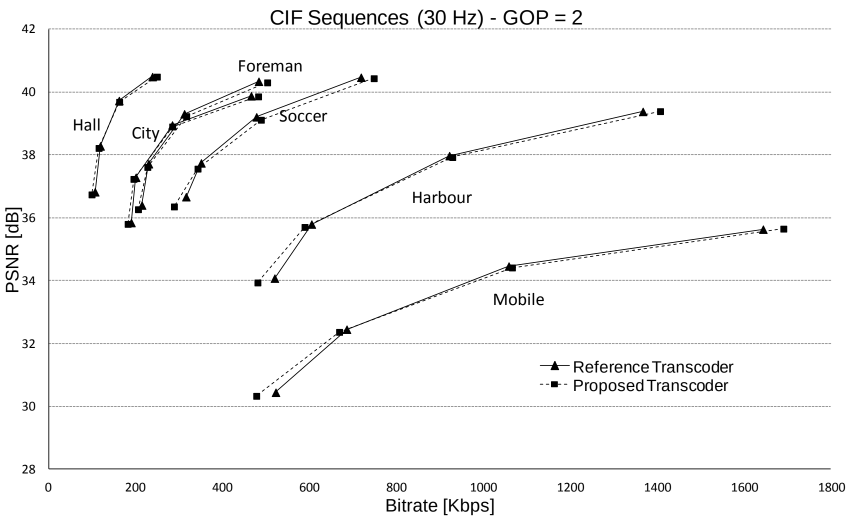
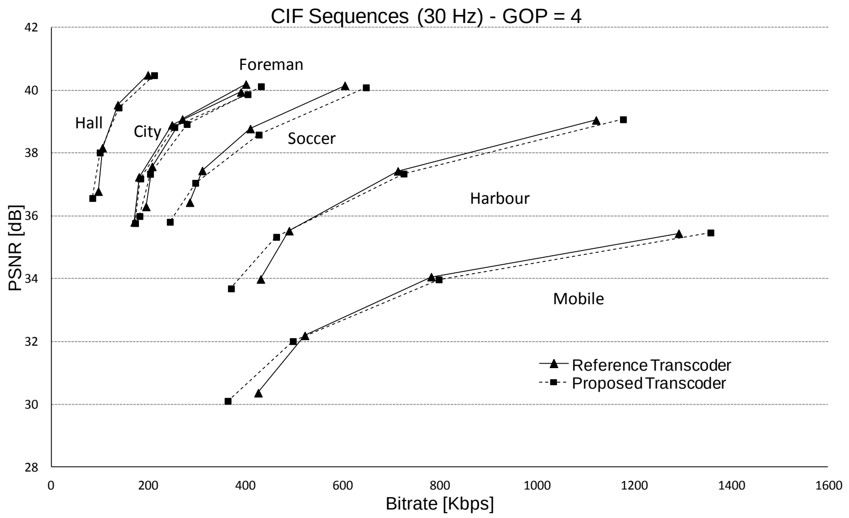


Figure 5.17. RD performance for the motion based transcoding in QCIF resolution with different GOP sizes - Main Profile





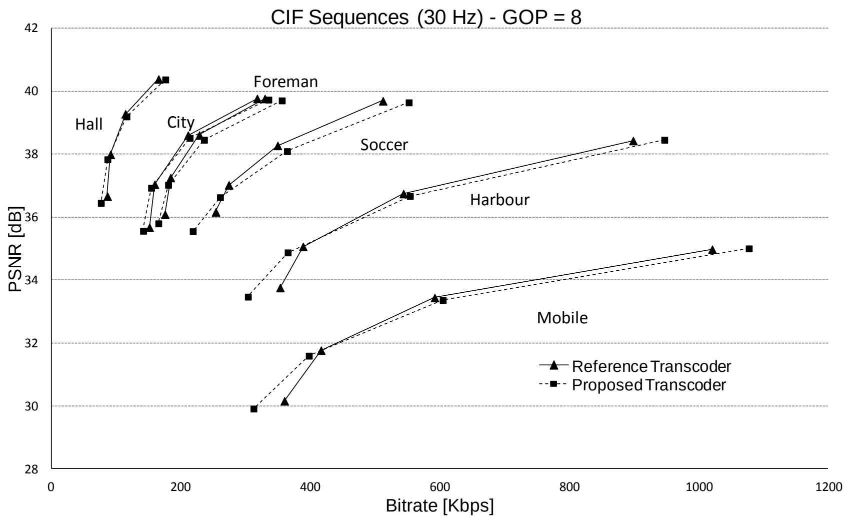
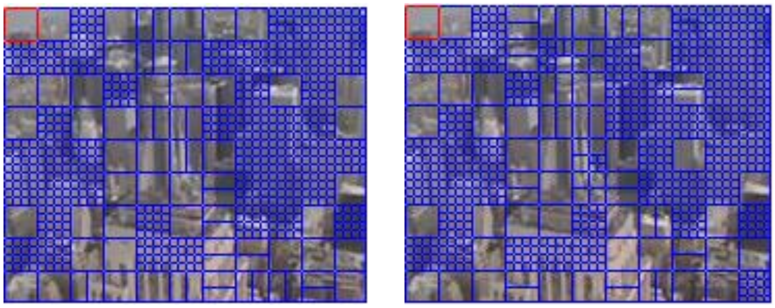
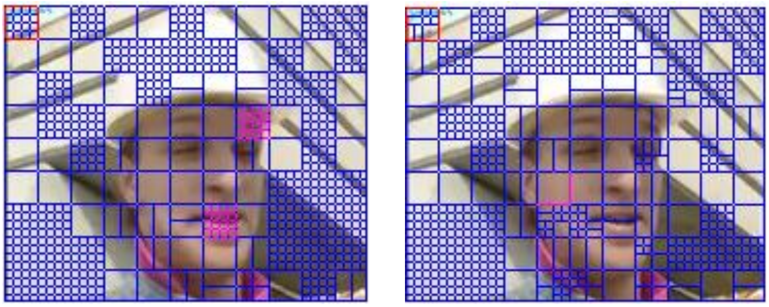


Figure 5.18. RD performance for the motion based transcoding in CIF resolution with different GOP sizes - Main Profile



(a) 2nd frame of City sequence (1st B-frame)



(b) 2nd frame of Foreman sequence (1st B-frame)

Figure 5.19. MB partitioning for the proposed H.264-to-SVC transcoder in Main Profile (left) compared to the reference one (right).

The performance results show as well that the algorithm works properly with different sequences with varying characteristics and resolutions, although there are some differences between sequences regarding the increment of bitrate. For example, the increment of bitrate is smaller in *Hall* or *Harbour* than in *Soccer*. This is due to the high movement of 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 reduced movement, the different prediction structure has less impact than if the sequence has high movement as the different.

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 the cases. The impact of the GOP size in the proposal is analyzed deeply in section 5.5.

5.4. Impact of Number of Temporal Layers

In this section, as in Section 4.5, is presented an analysis of how many temporal layers conforms the scenario that leads to a trade-off between reduction of coding complexity and maintenance of coding efficiency.

As commented previously, the fast MB mode decision algorithm has been applied to the two enhancement temporal layers because the encoder spent around an 80% of the encoding time in these layers. In this section, the results of some experiments run varying the number of temporal layers where the proposal was applied are shown.

5.4.1 Scenario and Metrics

Experiments were conducted to evaluate the impact of the number of temporal layer where is applied the proposal. To determine that, a GOP of 8 for QCIF resolutions and 16 for CIF resolutions, which implies to insert a frame of the base layer every 0.5s, is chosen. Then, several sequences were decoded and fully encoded for being used as reference and transcoded using the presented technique applied to different number of temporal layers to determine the optimal number of temporal layers where applying the algorithm.

The characteristics of the sequences and the conditions of the experiments are the same as in the previous performance evaluations (depicted in Section 4.3.2 for Baseline Profile and 4.3.3 for Main Profile).

The metrics used to evaluate the performance of the proposal are Time Saving (%), ΔBitrate (%) and ΔPSNR (dB). All these metrics have been defined previously in Section 4.3.4.

5.4.2 Results and Analysis

After decoding and re-encoding the sequences as reference transcoder and applying the algorithm presented to different combination of temporal layers, an average of Time Saving, ΔBitrate and ΔPSNR for every combination of temporal layers in Baseline Profile is calculated and represented in Figure 5.20 for Baseline Profile and for Main Profile in Figure 5.21.

The obtained results demonstrate that by applying the fast MB mode decision algorithm in different number of temporal layers, different results can be achieved, obtaining different RD performances as well as time saving just simply varying the temporal layers which the approach is applied. For example, for QCIF resolutions, the average of ΔBitrate varies from 1.2% to 6% depending on if the proposal is applied to one temporal layer or to four. The ΔPSNR varies from near 0.15 dB to about 0 dB. The Time Saving achieved goes from around 45% applying to one temporal layer to 75% if the technique is applied to four temporal layers.

As shown in Figure 5.20 and Figure 5.21, a trade-off between complexity reduction and RD penalty drop is achieved when the technique is applied to two temporal layers.

5.5. Impact of the GOP Size

Once it has been concluded that the optimal number of temporal layers where applying the proposal is two, another challenge is determining if the proposal is valid for various GOP sizes or if its behaviour varies widely depending of the GOP length.

5.5.1 Scenario and Metrics

For evaluating the impact of the GOP size in the proposal, several sequences were fully decoded and re-encoded for being used as reference and the same sequences were transcoded using the presented algorithm using different GOP length.

The characteristics of the sequences and the conditions of the experiments are the same as in the previous performance evaluations (depicted in Section 4.3.2 for Baseline Profile and 4.3.3 for Main Profile).

The metrics used to evaluate the performance of the proposal are Time Saving (%), ΔBitrate (%), and ΔPSNR (dB). All these metrics have been defined previously in Section 4.3.4.

5.5.2 Results and Analysis

The results obtained after run the codifications are shown in Section 5.3.2. The average of the Time saving, ΔBitrate and ΔPSNR results for every GOP sizes are represented in a graphical way in Figure 5.22 and Figure 5.23.

Both the tables and graphics can be seen that, although the values of ΔBitrate and ΔPSNR varies with the GOP, the reduction of time achieved in the whole sequence is always greater than 55% and reaches a 70% with a GOP size of 4. This variation is due to the technique presented is applied only to two enhancement temporal layers, but in the case of GOP length of 2, there is only one enhancement temporal layer, so the time reduction is smaller. However, when the transcoding technique is applied to sequences encoded with GOP size of 4, the time reduction achieves its maximum value, a 70%. This is due to in this case the technique is applied to two out of three temporal layers and only is encoded completely the temporal base layer. The partial time saving is constant (around an 84% for Baseline Profile and 87% for Main Profile). The ΔBitrate varies between less than 0.2% for a GOP size of 2 to 3% for a GOP size of 32). Regarding ΔPSNR varies from a gain of 0.2 dB to a loss of almost 0.10 dB.

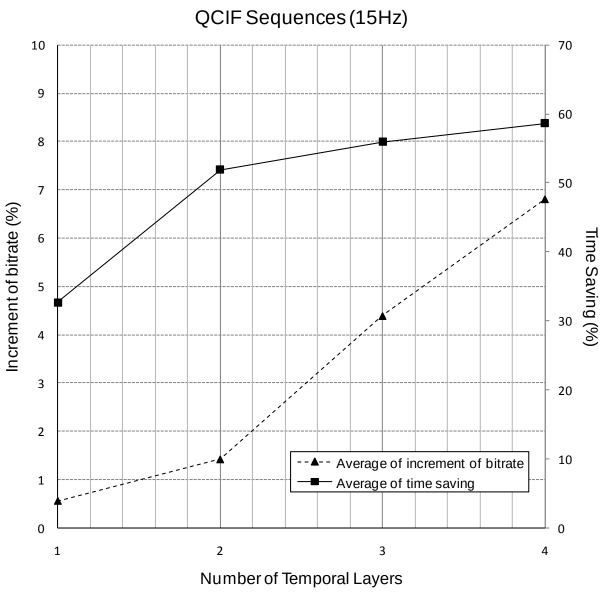
In conclusion, the proposal can be applied to different GOP sizes and works properly in all of them.

5.6 Conclusions

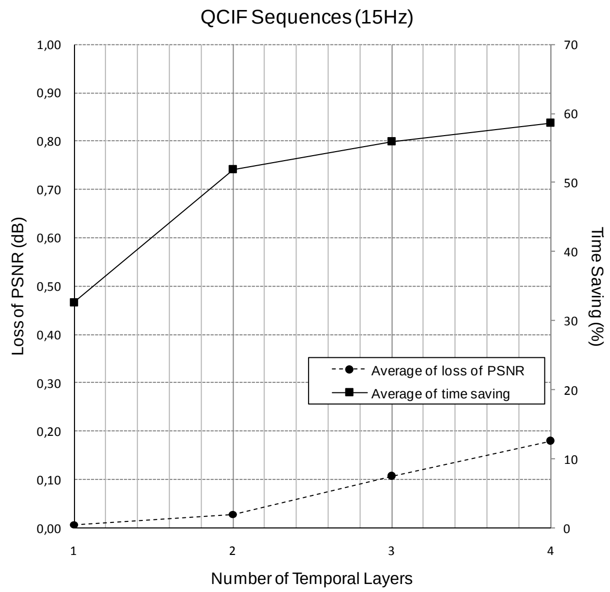
As it was said in the previous chapter, the reference transcoder decodes completely the video received and then encodes it to SVC. The most complex part of the transcoder is the encoder stage where the interprediction process takes up most of consuming resources. Focusing on the interprediction, the other task suitable to be accelerated, apart from ME, is the mode decision process.

In this chapter, as second contribution of this thesis, is presented an improved H.264/AVC-to-SVC transcoder that reduces the complexity around an 84% and 87% in the temporal layers where is applied in Baseline and Main Profile respectively. This improvement is achieved by choosing the MB types to be checked in the encoder stage depending on the information collected in the decoder stage. The specific MB types checked are selected by a decision tree built using ML tools.

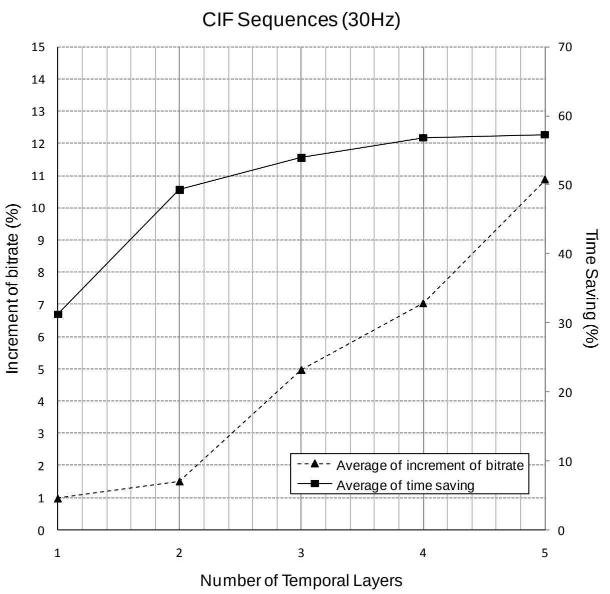
The experimental results presented in this chapter show that it is capable to reduce the coding complexity as it was said previously while maintaining the coding efficiency. Moreover, it is valid for different profiles, GOP sizes and resolutions



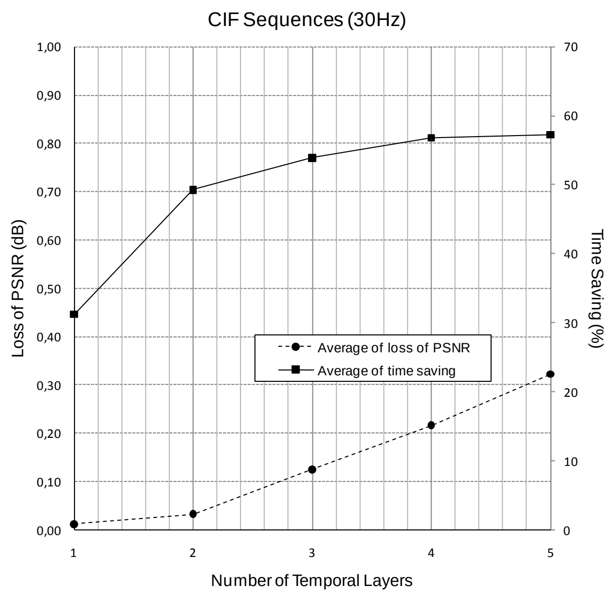
(a) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – QCIF resolution



(b) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – QCIF resolution

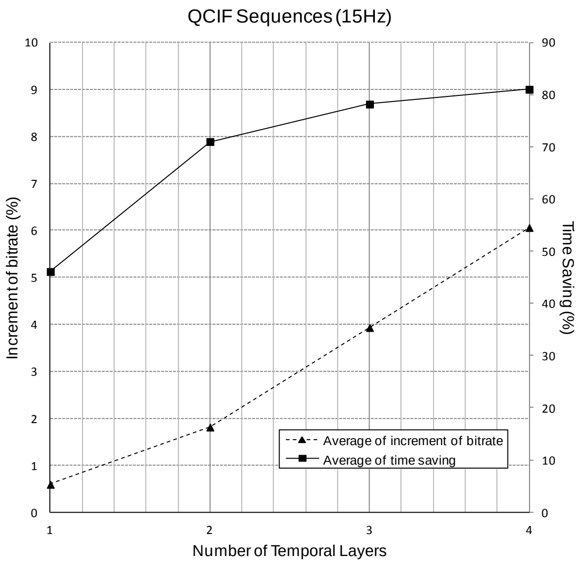


(c) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – CIF resolution

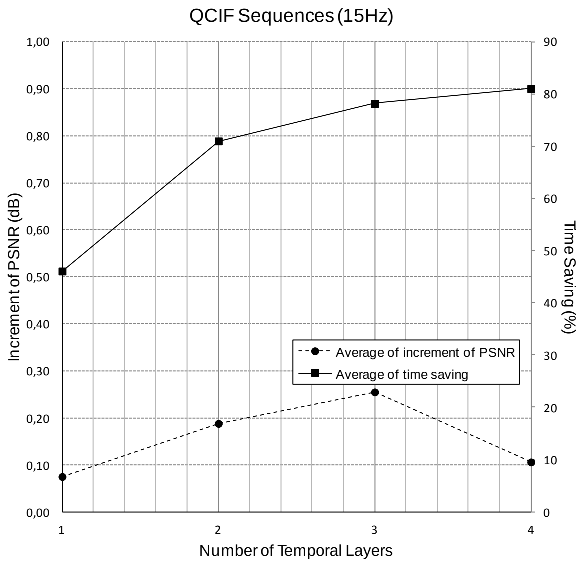


(d) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – CIF resolution

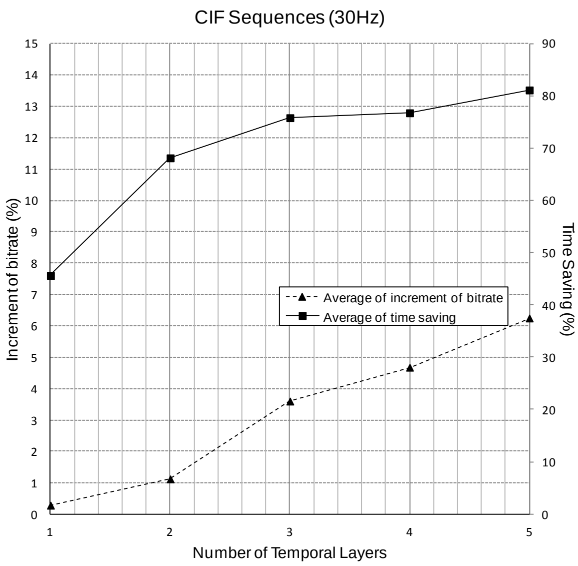
Figure 5.20. Average of increment of bitrate, loss of PSNR and time saving depending on the number of temporal layers transcoded for QCIF and CIF resolutions - Baseline Profile



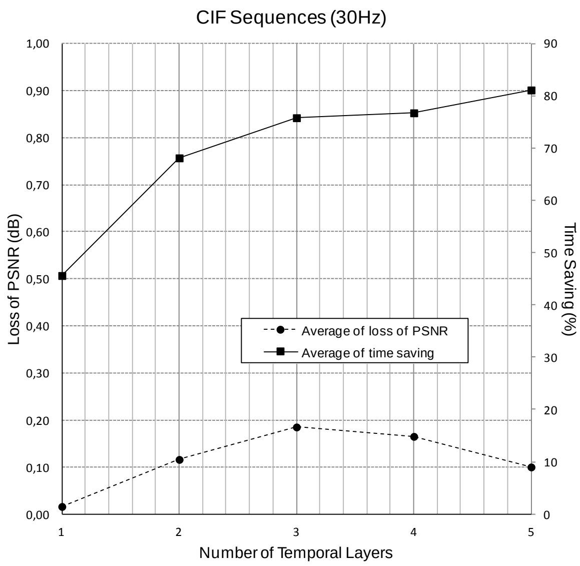
(a) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – QCIF resolution



(b) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – QCIF resolution

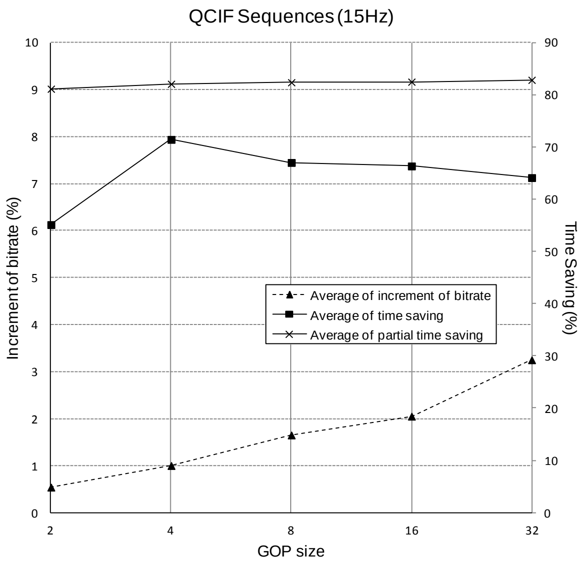


(c) ΔBitrate vs. Number of Temporal Layers vs. Time Saving – CIF resolution

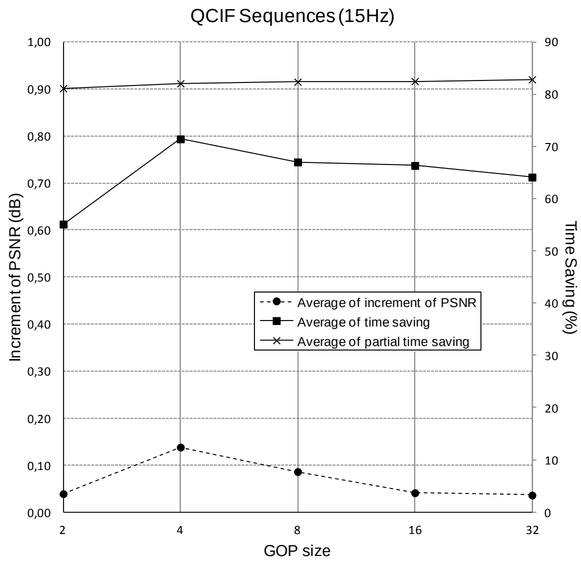


(d) Loss of PSNR vs. Number of Temporal Layers vs. Time Saving – CIF resolution

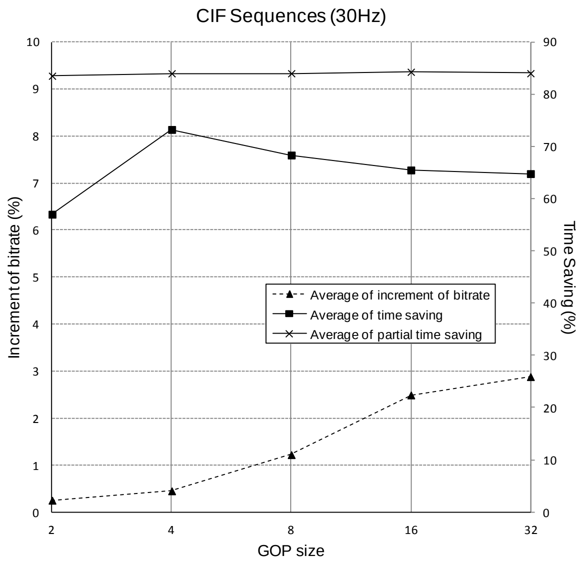
Figure 5.21. Average of increment of bitrate, loss of PSNR and time saving depending on the number of temporal layers transcoded for QCIF and CIF resolutions - Main Profile



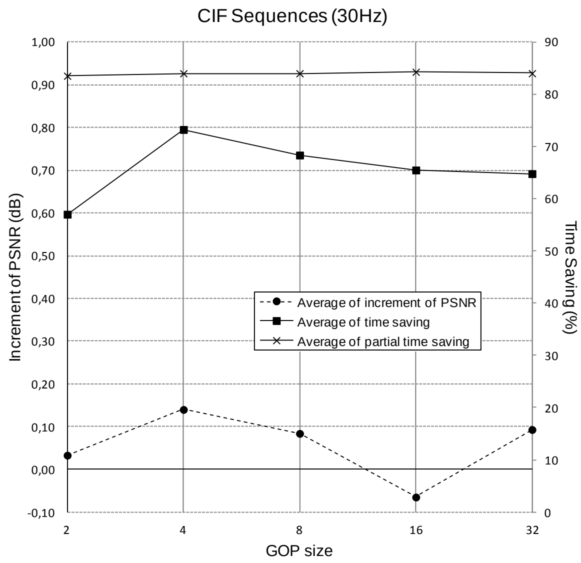
(a) ΔBitrate vs. GOP size vs. Time Saving – QCIF resolution



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

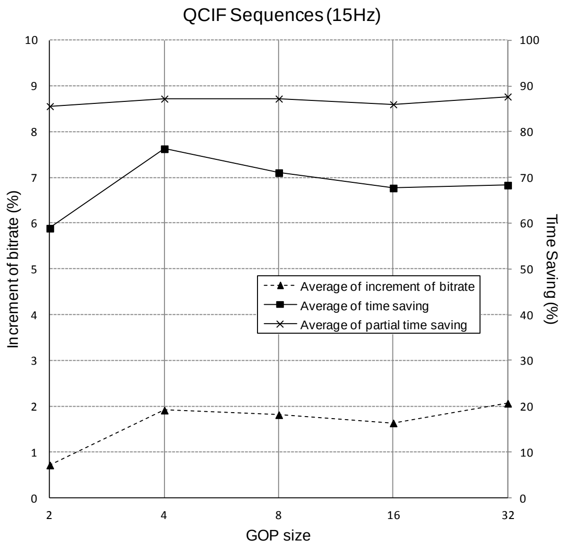


(c) ΔBitrate vs. GOP size vs. Time Saving – CIF resolution

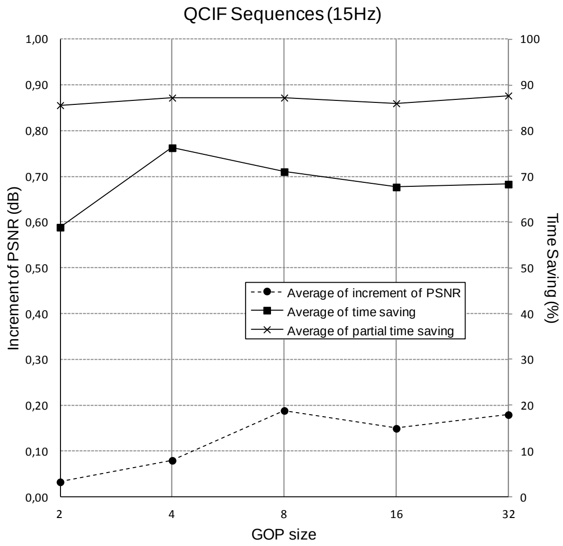


(d) Loss of PSNR vs. GOP size vs. Time Saving – CIF resolution

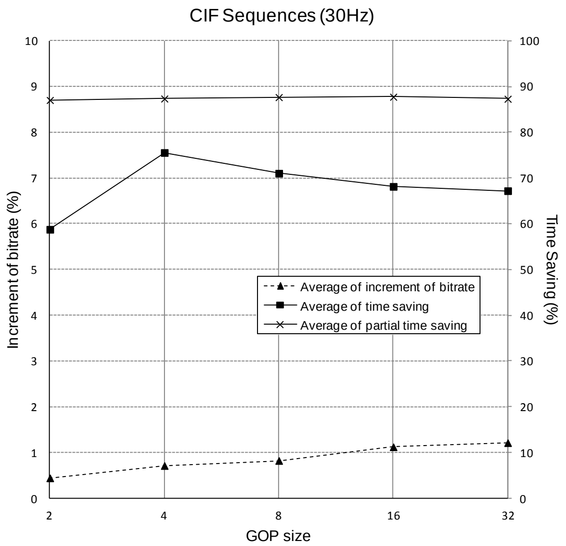
Figure 5.22. Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions - Baseline Profile



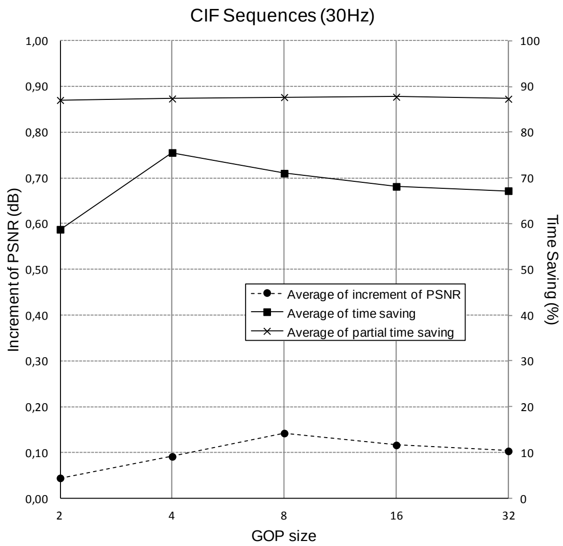
(a) ΔBitrate vs. GOP size vs. Time Saving – QCIF resolution



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



(c) ΔBitrate vs. GOP size vs. Time Saving – CIF resolution



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

Figure 5.23. Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions - Main Profile

CHAPTER 6

PROPOSED H.264/AVC-TO-SVC TRANSCODER

In this chapter, an H.264/AVC-to-SVC transcoder based on the techniques proposed in this thesis is presented.

6.1 Observations and Motivation

One of the key points that need to be addressed in the design of an efficient H.264/AVC-to-SVC transcoder is the Interprediction since it is one of computationally intensive task involved in the transcoding process. This chapter combines the Dynamic ME Window Approach proposed in chapter 4 and the Fast MB Mode Decision Algorithm based on ML techniques proposed in chapter 5 both in Baseline and Main Profile to be used as part of a H.264/AVC-to-SVC transcoder with temporal scalability.

6.2 Acceleration the Interprediction Process

In this chapter, as it said previously, the decision tree combined with the dynamic ME window approach is presented. The key point is to see if both mechanisms can work together efficiently and they form a transcoder capable of achieving a good trade-off between PSNR, bitrate and the time necessary to transcode the sequence.

The proposed transcoder consists of two parts: an H.264/AVC decoder followed by an SVC encoder. Firstly, the transcoder fully decodes the H.264/AVC sequence and then, the SVC encoder encodes the sequence using the information collected in the decoding stage. This information needed of each MB for running this proposal is the same as in the previous proposals. According to the answer given by the decision tree, the MB mode is adaptively selected during the SVC encoding process. Moreover, the H.264/AVC MVs are reused to dynamically reduce the search range window.

In Figure 6.1 is represented our proposed transcoder. The red parts denote the modules that have been modified and accelerated using some information incoming from the H.264/AVC decoder.

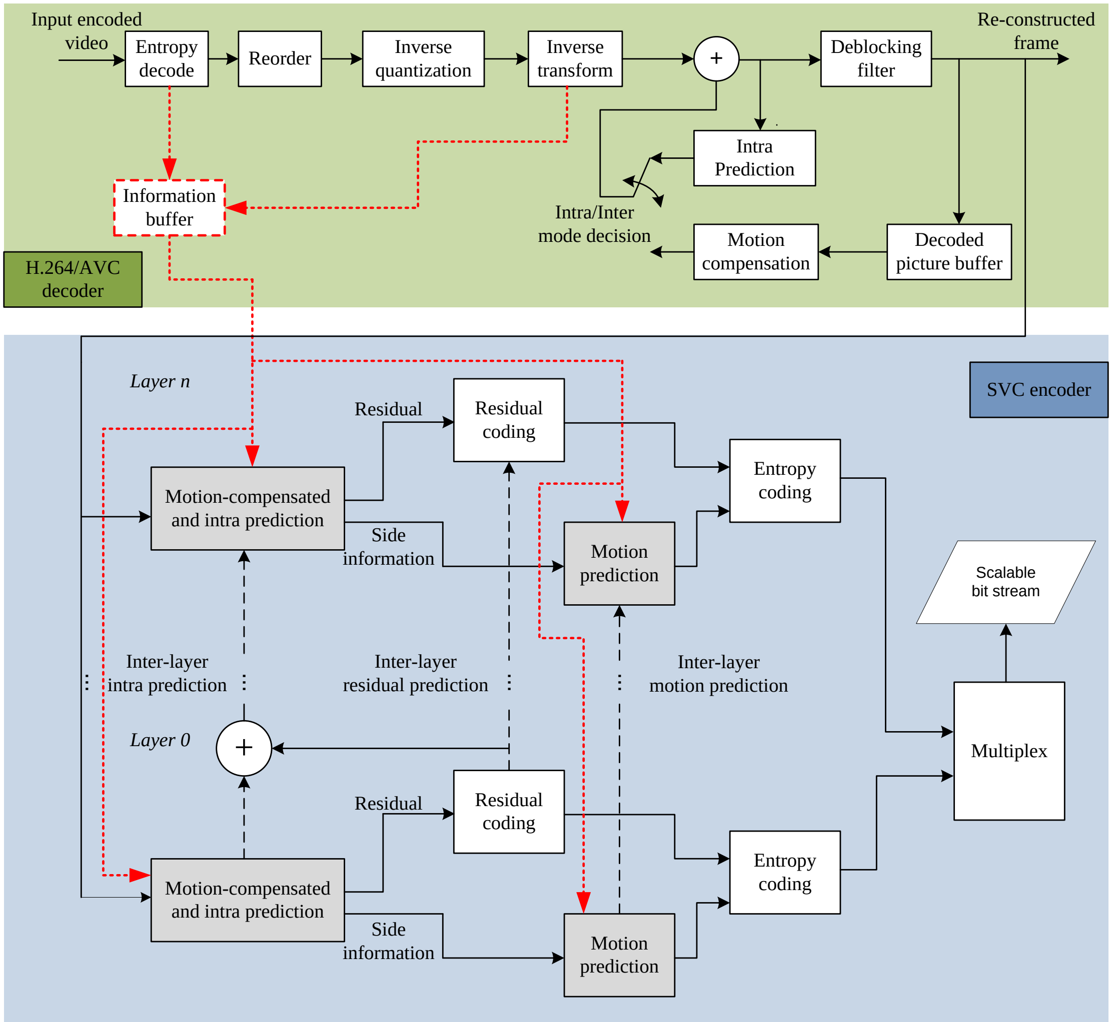


Figure 6.1. The proposed H.264/AVC-to-SVC transcoder

6.2.1 Baseline Profile Evaluation

This section discusses the performance evaluation of the proposed H.264/AVC-to-SVC transcoder for Baseline Profile [112].

For measuring the performance of the proposal, Time Saving, ΔPSNR, and ΔBitrate are obtained. These metrics as well as the encoding scenario are depicted in sections 4.3.2

and 4.3.4. As in the proposals presented in this thesis, the combined proposal was applied to the two enhancement temporal layers with highest identifier because a trade-off between time saving, bitrate increase and loss of PSNR were achieved when the separated proposals were applied to these temporal layers, so it is presumably that together the best option is to applied to them.

The results for running the technique for different GOP sizes are shown in Table 6.1-Table 6.5. Moreover, these results are collected graphically in Figure 6.4.

Some resulting RD curves for the SVC bitstreams with several GOP sizes are shown in Figure 6.2 and Figure 6.3 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.

Table 6.1. 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 – 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	66.10	98.85
City	0.026	0.91	66.88	99.04	0.056	0.21	66.00	98.96
Foreman	0.078	1.17	65.54	97.19	-0.058	1.45	65.21	97.67
Soccer	0.037	1.45	63.83	94.42	0.022	1.22	63.85	95.58
Harbour	0.027	-0.34	66.10	98.36	0.053	-0.57	64.15	96.95
Mobile	0.041	-0.40	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 6.2. 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 – 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	99.27
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***

Table 6.3. 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 – 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	0.159	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 6.4. 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 – 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	0.327	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***

Table 6.5. RD performance and time savings of the approach for GOP = 32 and different resolutions

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	75.60	92.09	0.757	1.11	74.94	98.22
City	-0.190	3.53	76.00	91.98	-0.101	2.68	75.01	98.44
Foreman	-0.110	5.35	73.77	96.53	-0.260	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	75.30	98.52	-0.005	2.31	74.76	97.88
Mobile	0.045	2.00	75.01	98.48	0.171	1.13	74.69	97.84
Average	***0.037***	***3.28***	***74.67***	***95.40***	***0.097***	***2.77***	***74.29***	***97.42***

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 a significant reduction of computational complexity is achieved (around a 98% where the proposal is applied).

6.2.2 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 process followed to obtain the results of Time Saving, ΔPSNR, and ΔBitrate is depicted in section 4.3.

As in the proposals presented in this Thesis, the combined proposal was applied to the two enhancement temporal layers with highest identifier because a trade-off between time saving, bitrate increase and loss of PSNR were achieved when the separated proposals were applied to these temporal layers, so it is presumably that together the best option is to applied to them.

The results for running the technique for different GOP sizes are shown in Table 6.6-Table 6.10. Moreover, these results are collected graphically in Figure 6.7. Some resulting RD curves for the SVC bitstreams with several GOP sizes are shown in Figure 6.5 and Figure 6.6 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.

6.2.3 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 during this thesis 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, as in chapter 5, 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.

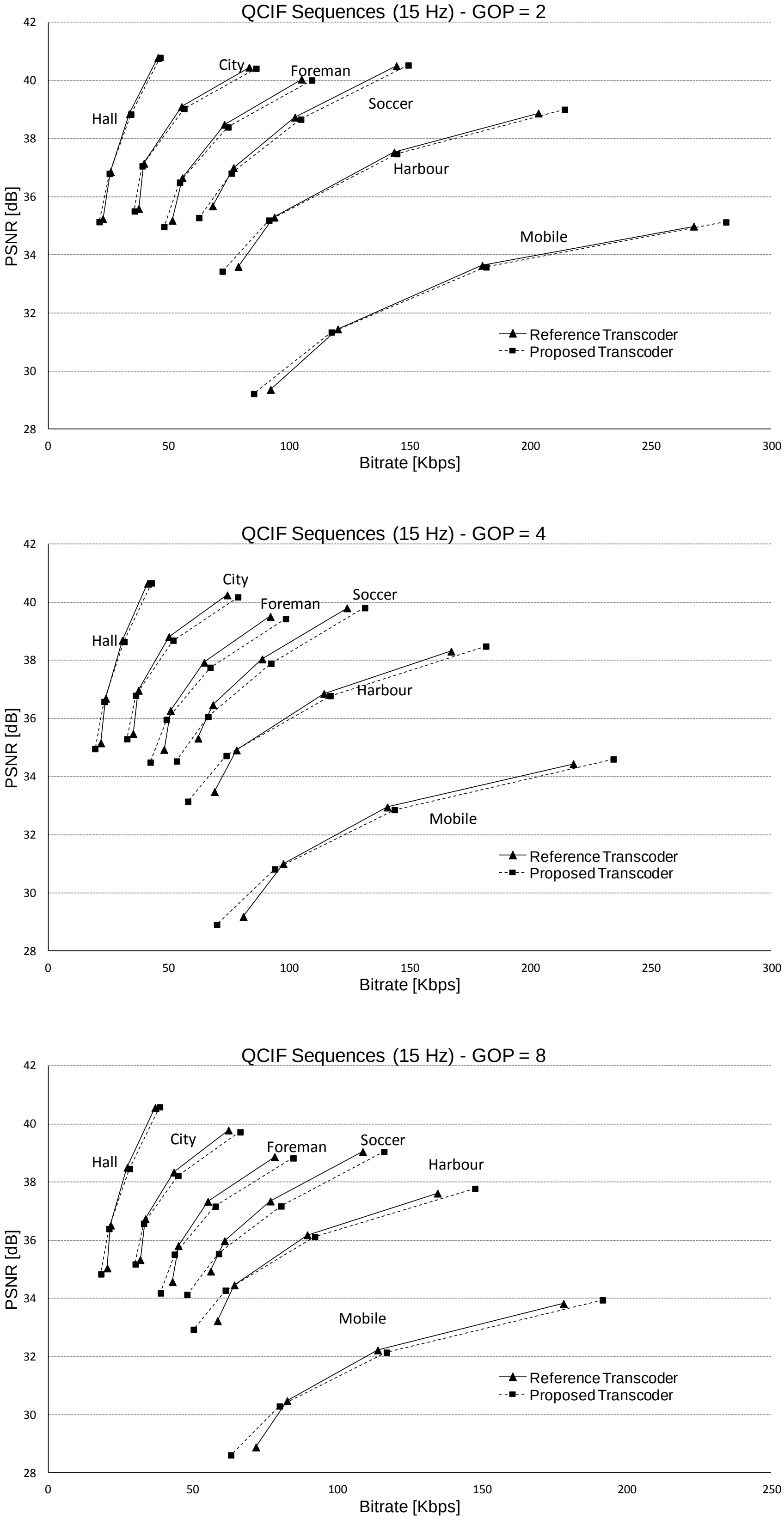


Figure 6.2. RD performance for the motion based transcoding in QCIF resolution with different GOP sizes – Baseline Profile

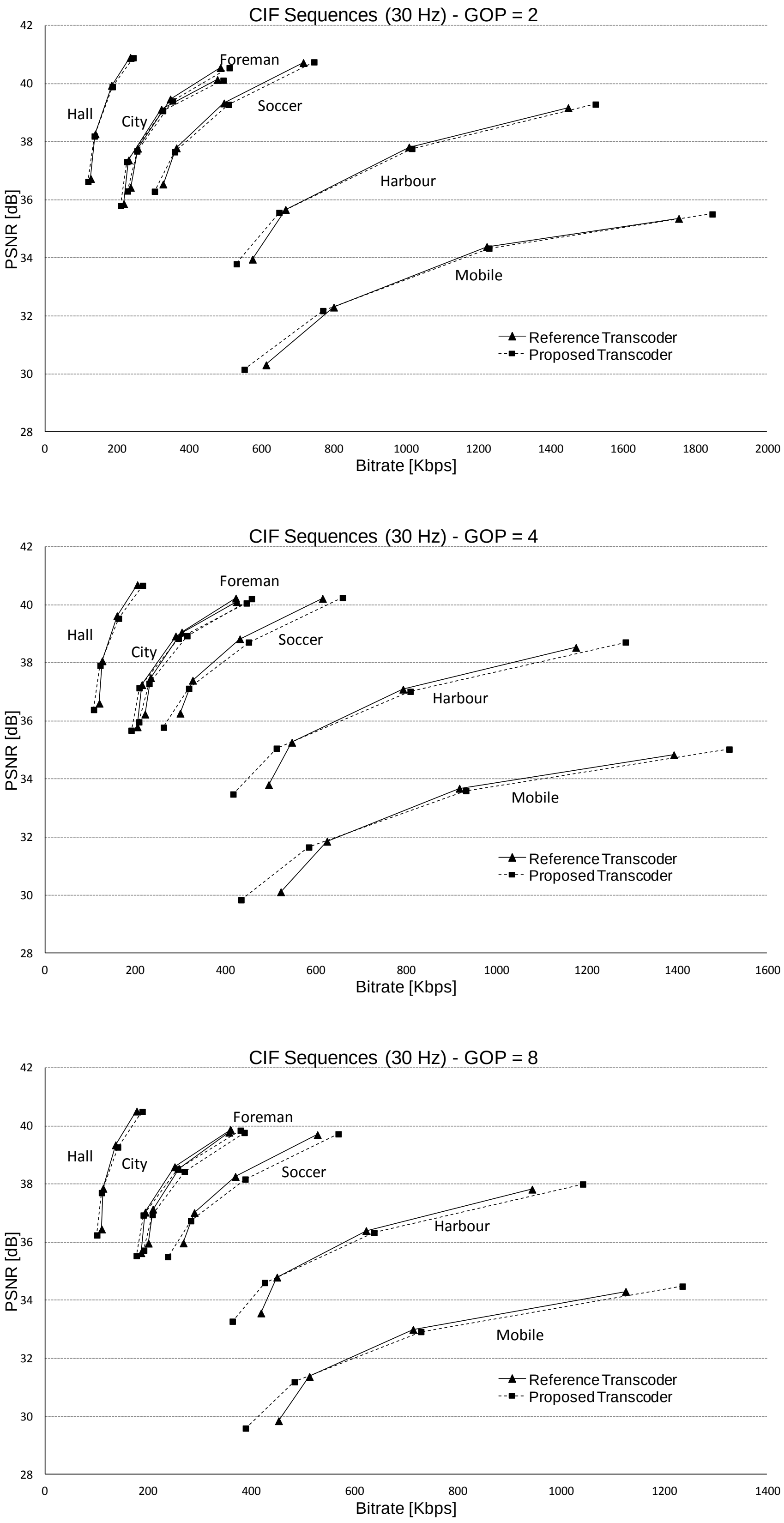
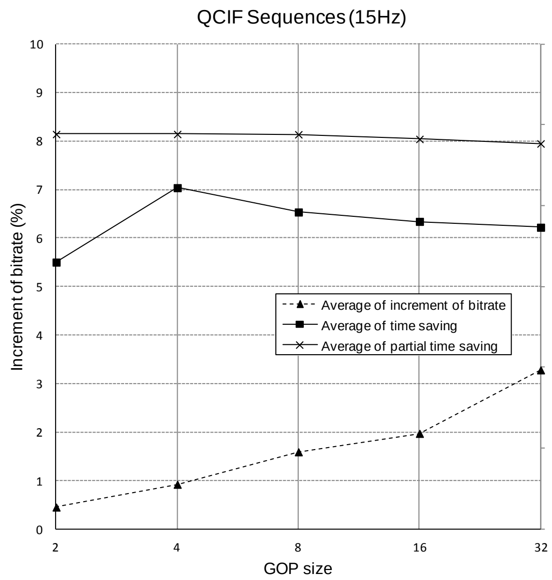
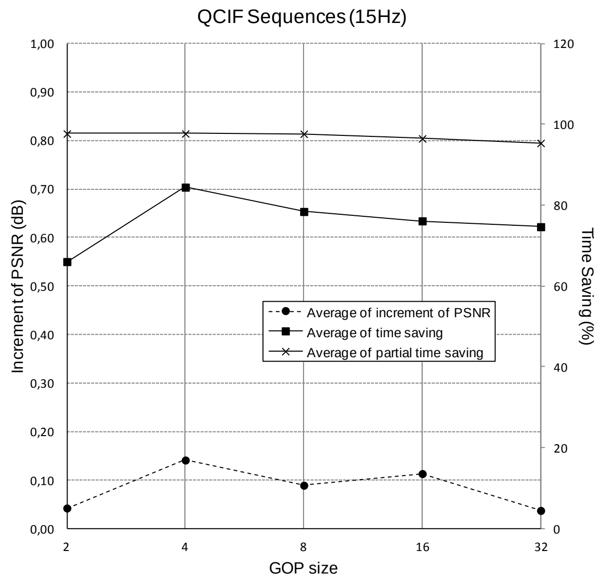


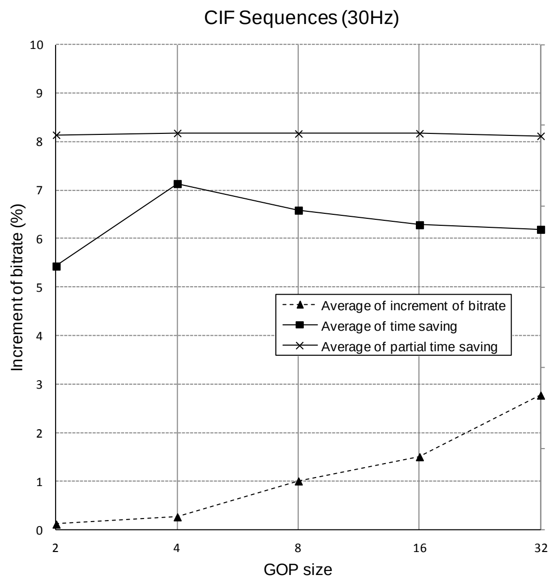
Figure 6.3. RD performance for the motion based transcoding in CIF resolution with different GOP sizes – Baseline Profile



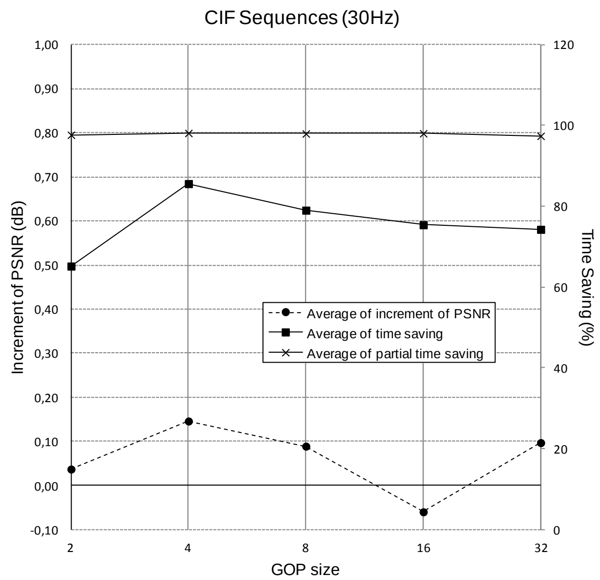
(a) ΔBitrate vs. GOP size vs. Time Saving – QCIF resolution



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



(c) ΔBitrate vs. GOP size vs. Time Saving – CIF resolution



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

Figure 6.4. Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions – Baseline Profile

Table 6.6. 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 – Main 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	0.065	-0.43	67.93	99.20	0.134	-1.18	65.63	99.13
City	0.060	0.34	68.80	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	-0.010	1.94	63.48	95.42
Harbour	0.070	-0.60	68.74	98.86	0.071	-0.55	65.51	98.78
Mobile	0.018	-0.20	68.80	99.01	0.060	-0.97	65.39	98.83
Average	***0.031***	***0.74***	***67.49***	***98.77***	***0.044***	***0.05***	***65.08***	***98.16***

Table 6.7. 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 – Main 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	0.168	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	-0.020	3.33	83.96	97.65
Soccer	-0.064	4.93	81.80	99.45	0.067	3.62	82.07	95.52
Harbour	0.175	-0.56	86.24	98.90	0.141	-1.01	85.00	98.83
Mobile	0.047	0.57	86.00	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***

Table 6.8. 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 – Main 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	0.666	-0.03	81.29	99.29	0.444	-1.12	79.34	99.16
City	0.065	1.75	79.89	99.11	0.020	0.23	79.31	99.20
Foreman	0.040	3.39	79.40	99.71	-0.175	3.15	78.14	97.68
Soccer	-0.016	4.97	75.72	99.43	0.094	4.10	76.38	95.55
Harbour	0.366	-0.66	80.29	98.90	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 6.9. 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 – Main 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	0.632	-0.06	76.12	97.80	0.337	-0.90	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	95.60
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***	***75.90***	***98.31***

Table 6.10. RD performance and time savings of the approach for GOP = 32 and different resolutions

RD performance and time savings of H.264/AVC-to-SVC transcoder GOP = 32 – Main 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	0.627	0.29	76.73	99.26	0.167	-0.34	74.88	98.57
City	-0.019	1.63	76.07	99.17	0.100	0.40	75.09	98.54
Foreman	-0.024	3.50	74.42	96.70	-0.002	3.40	74.02	97.03
Soccer	-0.013	5.23	71.40	94.44	0.076	4.38	72.37	95.00
Harbour	0.374	-0.02	76.30	98.96	0.231	-0.26	75.06	98.28
Mobile	0.107	0.86	75.71	99.16	0.150	0.19	75.75	98.58
Average	***0.175***	***1.92***	***75.11***	***97.95***	***0.120***	***1.30***	***74.53***	***97.67***

6.3 Overall Performance Evaluation

In this section a comparison between the results presented in this thesis is done.

6.3.1 Baseline Profile

From Table 6.11 to Table 6.15 are summarized the results presented during this thesis for Baseline Profile. The proposal denoted by DMEW was presented in Chapter 4, the one denoted by DT was presented in Chapter 5 and then, the one denoted by DMEW+DT is the joint proposal presented in this chapter.

6.3.2 Main Profile

From Table 6.16 to Table 6.20 are summarized the results presented during this thesis for Main Profile.

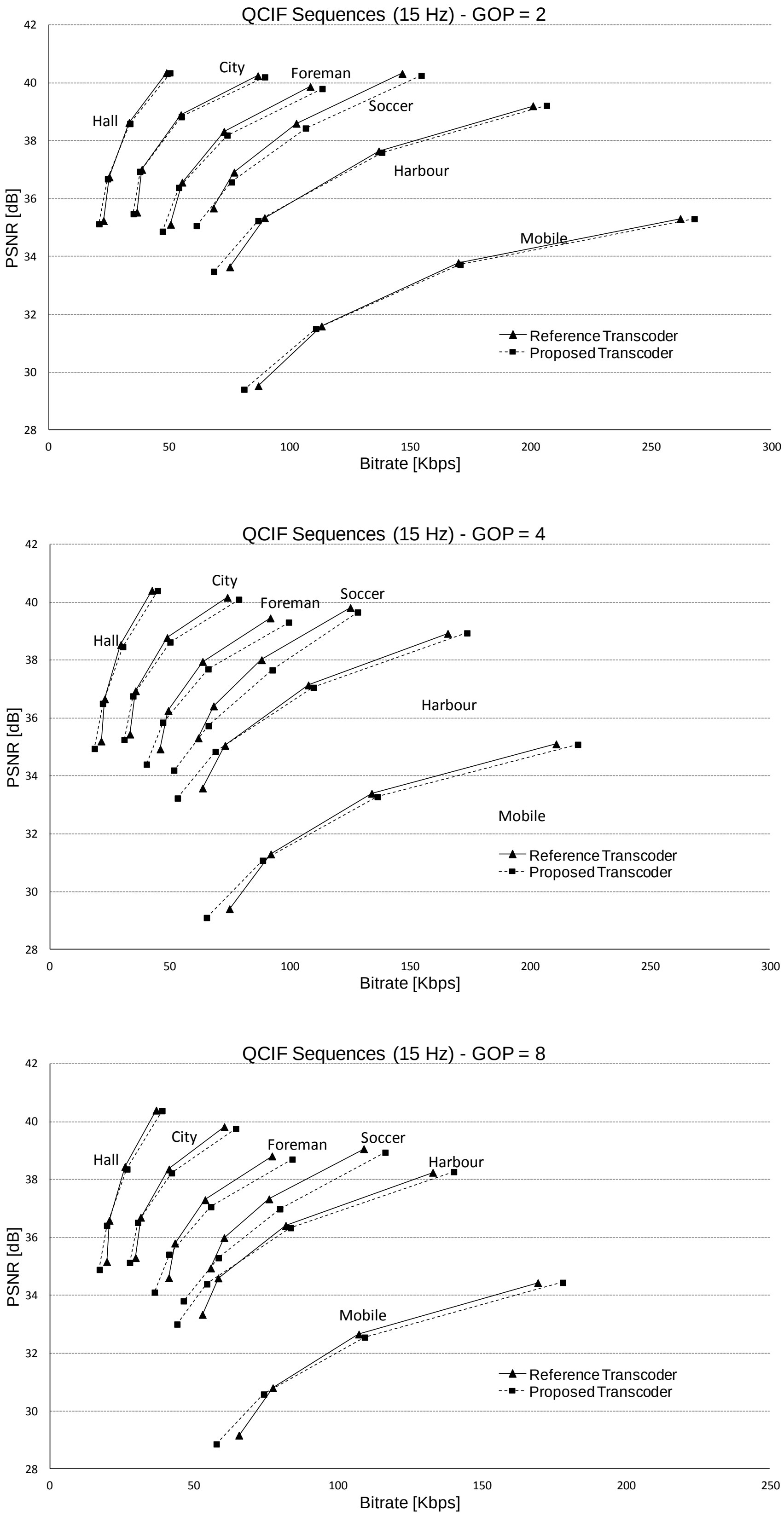


Figure 6.5. RD performance for the motion based transcoding in QCIF resolution with different GOP sizes – Main Profile

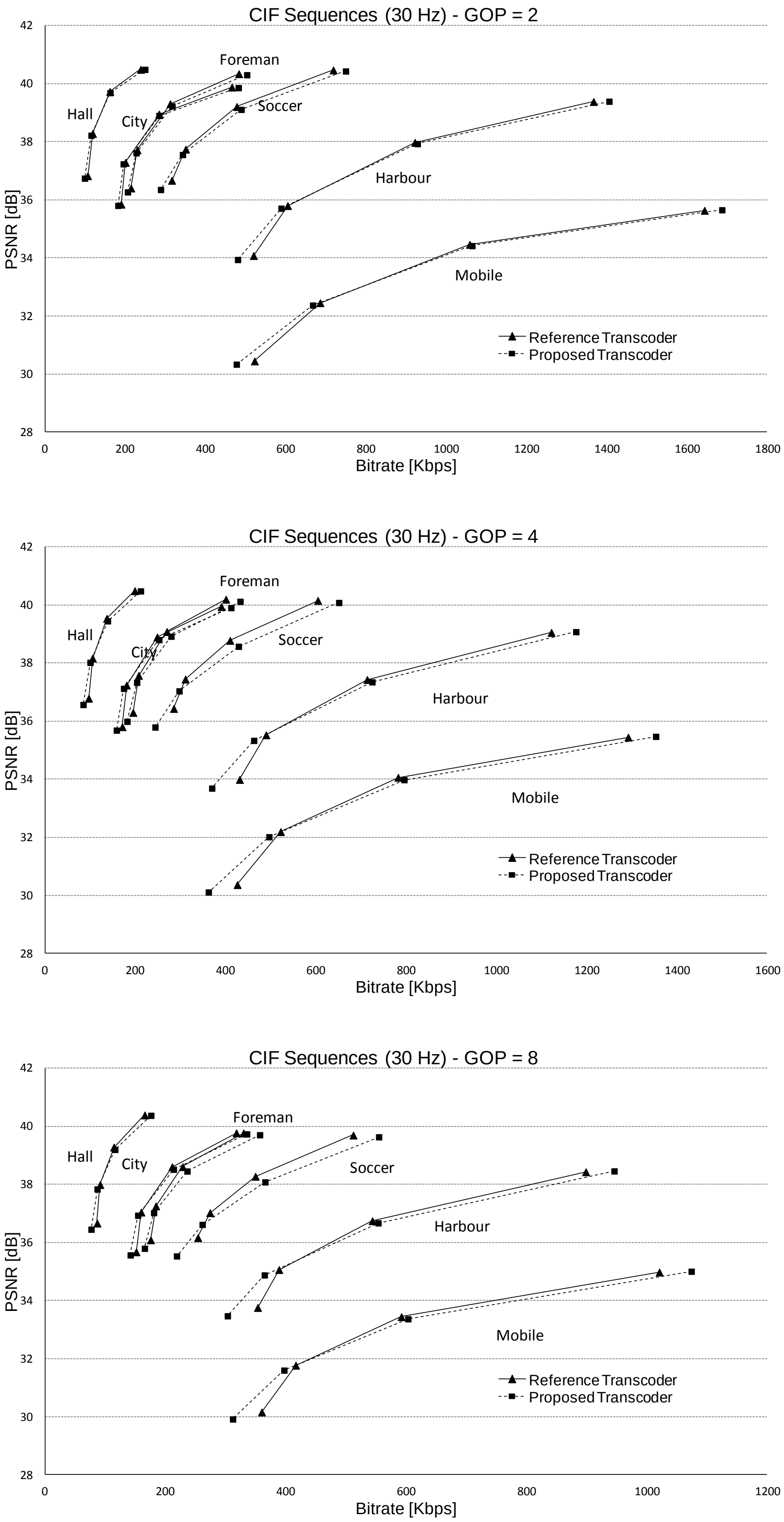
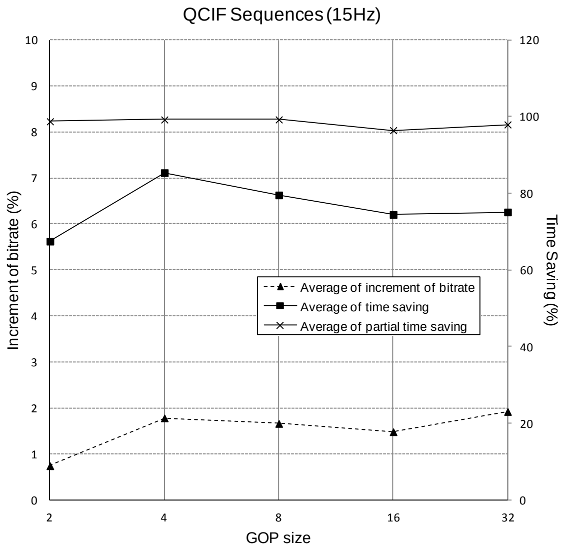
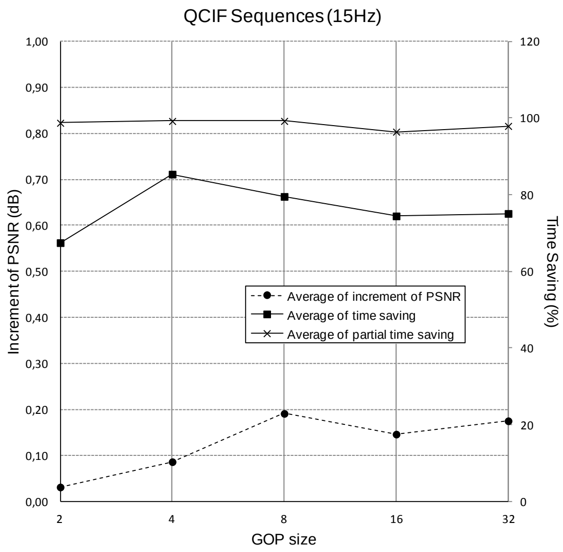


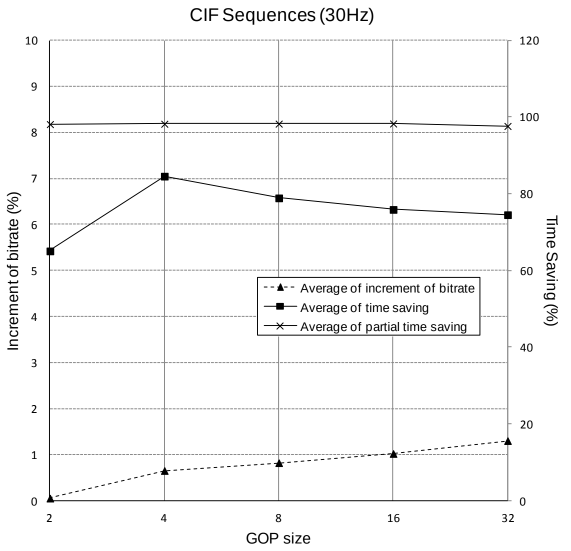
Figure 6.6. RD performance for the motion based transcoding in CIF resolution with different GOP sizes – Main Profile



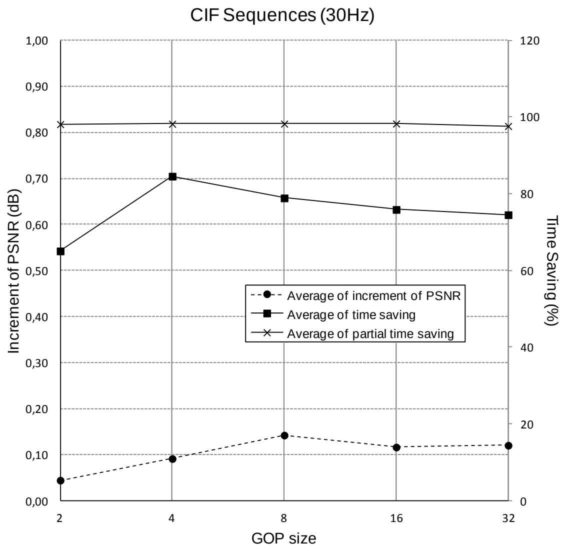
(a) ΔBitrate vs. GOP size vs. Time Saving – QCIF resolution



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



(c) ΔBitrate vs. GOP size vs. Time Saving – CIF resolution



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

Figure 6.7. Average of increment of bitrate, loss of PSNR and time saving depending on the GOP size for QCIF and CIF resolutions – Main Profile

As in Baseline Profile, the proposal denoted by DMEW was presented in Chapter 4, the one denoted by DT was presented in Chapter 5 and then, the one denoted by DMEW+DT is the joint proposal presented in this chapter.

Table 6.11. Comparison of the different proposals within this thesis - Baseline Profile and GOP = 2

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	-0.004	0.16	61.50	91.04	0.000	0.05	57.63	86.43
	DT	0.042	-0.05	57.96	85.38	0.055	-0.08	58.94	86.64
	DMEW+DT	***0.044***	***-0.11***	***66.94***	***99.12***	***0.055***	***-0.11***	***66.10***	***98.85***
City	DMEW	-0.006	0.62	55.35	82.01	-0.006	0.52	41.79	63.45
	DT	0.026	0.92	57.16	84.16	0.055	0.25	58.24	85.61
	DMEW+DT	***0.026***	***0.91***	***66.88***	***99.04***	***0.056***	***0.21***	***66.00***	***98.96***
Foreman	DMEW	-0.009	0.37	42.92	63.71	-0.010	0.52	41.79	63.45
	DT	0.077	1.21	56.20	82.70	-0.059	1.51	58.12	85.46
	DMEW+DT	***0.078***	***1.17***	***65.54***	***97.19***	***-0.058***	***1.45***	***65.21***	***97.67***
Soccer	DMEW	-0.079	2.92	31.26	46.41	-0.069	2.94	35.82	43.27
	DT	0.036	1.45	54.34	79.86	0.021	1.28	56.28	82.85
	DMEW+DT	***0.037***	***1.45***	***63.83***	***94.42***	***0.022***	***1.22***	***63.85***	***95.58***
Harbour	DMEW	0.007	-0.04	60.68	89.97	0.003	-0.15	60.09	87.26
	DT	0.022	-0.13	52.91	77.95	0.047	-0.35	56.12	80.58
	DMEW+DT	***0.027***	***-0.34***	***66.10***	***98.36***	***0.053***	***-0.57***	***64.15***	***96.95***
Mobile	DMEW	0.003	-0.09	58.83	87.13	0.004	-0.12	56.19	84.74
	DT	0.033	-0.15	52.28	76.93	0.080	-1.10	54.51	80.09
	DMEW+DT	***0.041***	***-0.40***	***66.97***	***98.26***	***0.092***	***-1.55***	***65.81***	***98.15***

Table 6.12. Comparison of the different proposals within this thesis - Baseline Profile and GOP = 4

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	-0.001	0.40	75.64	87.41	0.000	0.19	77.49	89.43
	DT	0.219	0.04	74.58	85.80	0.328	-0.45	74.69	86.45
	DMEW+DT	***0.222***	***-0.02***	***85.91***	***99.14***	***0.331***	***-0.53***	***86.64***	***99.08***
City	DMEW	-0.055	1.66	67.27	77.73	-0.068	3.33	63.57	73.38
	DT	0.064	1.93	75.69	86.04	0.200	0.66	76.30	86.96
	DMEW+DT	***0.066***	***1.87***	***86.13***	***99.11***	***0.204***	***0.59***	***87.23***	***99.27***
Foreman	DMEW	-0.006	0.81	50.72	58.76	-0.026	1.19	51.51	59.54
	DT	0.251	2.34	72.68	83.55	-0.112	3.01	74.63	85.65
	DMEW+DT	***0.259***	***2.16***	***83.25***	***97.01***	***-0.108***	***2.92***	***84.64***	***97.70***
Soccer	DMEW	-0.093	3.94	36.56	42.43	-0.126	4.92	38.88	44.99
	DT	0.043	2.24	72.11	81.83	0.021	2.37	72.35	83.05
	DMEW+DT	***0.037***	***2.51***	***81.77***	***94.52***	***0.022***	***2.30***	***82.58***	***95.60***
Harbour	DMEW	0.008	0.13	74.67	86.28	0.020	0.01	75.04	86.54
	DT	0.107	-0.68	68.30	78.88	0.175	-1.22	71.75	81.57
	DMEW+DT	***0.112***	***-0.82***	***85.38***	***98.44***	***0.181***	***-1.43***	***87.30***	***98.87***
Mobile	DMEW	-0.023	0.99	71.72	82.79	-0.018	0.89	69.77	80.45
	DT	0.142	0.15	65.37	76.51	0.229	-1.69	69.83	80.37
	DMEW+DT	***0.151***	***-0.17***	***84.33***	***98.19***	***0.246***	***-2.30***	***85.24***	***98.24***

Table 6.13. Comparison of the different proposals within this thesis - Baseline Profile and GOP = 8

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	-0.011	0.59	70.51	87.46	-0.003	0.26	68.39	85.57
	DT	0.158	0.37	70.59	86.28	0.025	0.47	70.69	86.83
	DMEW+DT	***0.159***	***0.35***	***80.00***	***98.74***	***0.026***	***0.45***	***79.84***	***98.87***
City	DMEW	-0.075	2.06	78.50	78.50	-0.051	3.61	58.61	66.91
	DT	-0.008	2.67	70.16	85.70	0.175	1.32	70.10	86.16
	DMEW+DT	***-0.003***	***2.61***	***80.02***	***99.12***	***0.178***	***1.25***	***79.83***	***99.04***
Foreman	DMEW	0.015	0.75	45.19	56.27	-0.056	1.48	43.22	53.93
	DT	0.210	3.22	66.89	82.89	-0.001	3.58	69.96	85.91
	DMEW+DT	***0.219***	***3.03***	***77.29***	***96.89***	***0.005***	***3.48***	***78.78***	***97.73***
Soccer	DMEW	-0.076	4.48	33.40	41.92	-0.105	4.97	36.10	42.45
	DT	0.074	2.61	65.19	80.63	-0.001	2.99	68.07	83.55
	DMEW+DT	***0.066***	***2.96***	***75.45***	***94.49***	***0.000***	***2.55***	***76.96***	***95.67***
Harbour	DMEW	0.005	-0.18	69.61	86.29	-0.005	0.25	69.66	86.47
	DT	0.048	0.15	64.60	79.54	0.072	-0.18	65.54	80.60
	DMEW+DT	***0.052***	***0.02***	***78.63***	***98.40***	***0.077***	***-0.38***	***79.46***	***98.37***
Mobile	DMEW	-0.020	0.90	66.68	82.77	-0.018	0.98	62.73	82.90
	DT	0.031	0.87	64.82	79.36	0.233	-0.84	65.81	81.10
	DMEW+DT	***0.038***	***0.57***	***79.24***	***98.36***	***0.248***	***-1.37***	***79.31***	***98.34***

Table 6.14. Comparison of the different proposals within this thesis - Baseline Profile and GOP = 16

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	-0.012	0.51	67.48	86.62	0.066	0.42	66.51	86.09
	DT	0.325	0.58	69.47	85.97	-0.673	1.90	67.61	86.89
	DMEW+DT	***0.327***	***0.52***	***77.63***	***97.86***	***-0.671***	***1.65***	***76.17***	***98.99***
City	DMEW	-0.167	3.21	60.59	77.79	-0.181	3.28	56.98	73.81
	DT	-0.040	3.14	69.01	85.45	-0.140	1.96	67.16	86.39
	DMEW+DT	***-0.035***	***3.06***	***77.54***	***97.87***	***-0.138***	***1.90***	***76.22***	***99.10***
Foreman	DMEW	0.023	0.89	44.76	57.66	-0.049	1.20	44.89	58.53
	DT	-0.333	3.36	65.30	82.47	-0.104	4.86	66.74	85.88
	DMEW+DT	***0.088***	***3.12***	***73.98***	***95.49***	***-0.097***	***4.78***	***75.06***	***97.65***
Soccer	DMEW	-0.107	5.02	32.65	42.30	-0.101	4.76	33.49	43.88
	DT	0.068	3.03	66.02	81.60	0.031	3.60	65.29	83.83
	DMEW+DT	***0.063***	***3.32***	***73.81***	***93.55***	***0.032***	***3.66***	***73.43***	***95.71***
Harbour	DMEW	0.040	0.22	66.69	85.51	0.135	-3.41	66.19	86.12
	DT	0.199	0.99	65.13	80.51	0.280	2.43	62.78	81.15
	DMEW+DT	***0.204***	***0.86***	***77.34***	***97.27***	***0.285***	***-2.60***	***75.68***	***98.46***
Mobile	DMEW	-0.029	1.40	63.92	82.00	-0.021	1.56	61.73	80.03
	DT	0.024	1.18	63.31	79.07	0.218	0.17	63.41	81.66
	DMEW+DT	***0.030***	***0.91***	***76.41***	***97.07***	***0.232***	***-0.39***	***75.93***	***98.45***

Table 6.15. Comparison of the different proposals within this thesis - Baseline Profile and GOP = 32

		QCIF (15 Hz)				***CIF (30 Hz)***			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	0.004	0.54	66.85	87.34	0.256	0.51	65.43	86.09
	DT	0.291	1.16	66.84	86.94	0.756	1.17	66.62	86.39
	DMEW+DT	***0.293***	***1.16***	***75.60***	***92.09***	***0.757***	***1.11***	***74.94***	***98.22***
City	DMEW	-0.111	2.41	60.84	79.25	-0.178	5.12	56.09	73.89
	DT	-0.192	3.64	66.89	85.93	-0.104	2.77	66.27	85.99
	DMEW+DT	***-0.190***	***3.53***	***76.00***	***91.98***	***-0.101***	***2.68***	***75.01***	***98.44***
Foreman	DMEW	-0.035	0.84	41.21	53.96	-0.049	1.20	44.89	58.53
	DT	-0.116	5.51	63.75	82.56	-0.264	5.31	66.05	85.59
	DMEW+DT	***-0.110***	***5.35***	***73.77***	***96.53***	***-0.260***	***5.17***	***73.96***	***97.07***
Soccer	DMEW	-0.136	5.87	35.11	45.72	-0.132	5.03	33.47	44.63
	DT	0.073	4.53	63.92	81.96	0.019	3.97	64.59	83.52
	DMEW+DT	***0.057***	***5.35***	***72.34***	***94.82***	***0.017***	***4.23***	***72.38***	***95.05***
Harbour	DMEW	0.047	0.27	66.38	86.30	0.059	0.21	65.40	86.11
	DT	0.122	2.41	61.86	80.05	-0.009	2.46	62.64	81.25
	DMEW+DT	***0.126***	***2.29***	***75.30***	***98.52***	***-0.005***	***2.31***	***74.76***	***97.88***
Mobile	DMEW	-0.095	2.74	63.60	82.95	-0.014	1.58	60.74	80.06
	DT	0.039	2.25	61.59	79.70	0.158	1.62	62.65	81.46
	DMEW+DT	***0.045***	***2.00***	***75.01***	***98.48***	***0.171***	***1.13***	***74.69***	***97.84***

Table 6.16. Comparison of the different proposals within this thesis - Main Profile and GOP = 2

		QCIF (15 Hz)				***CIF (30 Hz)***			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	0.007	0.10	64.00	90.38	0.003	0.06	56.50	85.70
	DT	0.064	-0.39	58.13	87.06	0.069	-0.48	58.00	85.84
	DMEW+DT	***0.065***	***-0.43***	***67.93***	***99.20***	***0.134***	***-1.18***	***65.63***	***99.13***
City	DMEW	0.004	0.28	45.18	65.34	-0.006	0.31	42.28	64.58
	DT	0.061	0.34	59.70	87.92	0.049	-0.41	59.39	88.15
	DMEW+DT	***0.060***	***0.34***	***68.80***	***99.12***	***0.049***	***-0.44***	***65.61***	***99.18***
Foreman	DMEW	-0.008	0.37	41.83	58.62	-0.007	0.55	40.70	57.90
	DT	-0.004	1.51	57.17	86.76	-0.039	1.42	59.74	88.30
	DMEW+DT	***-0.005***	***1.57***	***65.63***	***96.99***	***-0.041***	***1.52***	***64.84***	***97.62***
Soccer	DMEW	-0.061	2.43	29.67	42.19	-0.055	2.79	26.55	40.76
	DT	-0.016	3.46	62.28	81.24	0.133	1.16	59.36	88.10
	DMEW+DT	***-0.023***	***3.75***	***65.05***	***99.45***	***-0.010***	***1.94***	***63.48***	***95.42***
Harbour	DMEW	0.003	0.05	59.89	86.96	0.003	-0.07	55.44	83.87
	DT	0.068	-0.52	58.18	85.36	0.055	-0.77	57.20	84.90
	DMEW+DT	***0.070***	***-0.60***	***68.74***	***98.86***	***0.071***	***-0.55***	***65.51***	***98.78***
Mobile	DMEW	0.000	0.03	62.67	88.14	0.002	-0.05	54.98	83.43
	DT	0.016	-0.13	57.83	85.05	-0.005	1.71	58.89	86.90
	DMEW+DT	***0.018***	***-0.20***	***68.80***	***99.01***	***0.060***	***-0.97***	***65.39***	***98.83***

Table 6.17. Comparison of the different proposals within this thesis - Main Profile and GOP = 4

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	0.016	0.60	77.64	88.33	-0.004	0.45	77.69	89.43
	DT	0.166	0.15	77.92	88.55	0.230	-1.30	76.24	88.17
	DMEW+DT	***0.168***	***0.12***	***87.02***	***99.27***	***0.231***	***-1.33***	***85.27***	***99.15***
City	DMEW	-0.039	1.38	55.08	63.40	-0.126	3.26	59.61	67.90
	DT	0.141	2.07	77.47	88.49	0.017	0.36	76.29	88.23
	DMEW+DT	***0.143***	***2.03***	***86.47***	***99.12***	***0.019***	***0.26***	***85.21***	***99.19***
Foreman	DMEW	-0.028	1.08	48.07	55.26	-0.041	1.47	53.34	60.72
	DT	0.048	3.46	77.21	88.08	-0.016	3.20	76.51	88.35
	DMEW+DT	***0.046***	***3.57***	***84.67***	***99.71***	***-0.020***	***3.33***	***83.96***	***97.65***
Soccer	DMEW	-0.111	4.23	35.12	40.41	-0.135	5.64	36.81	42.65
	DT	-0.095	5.42	74.86	85.97	0.079	3.45	75.47	87.06
	DMEW+DT	***-0.064***	***4.93***	***81.80***	***99.45***	***0.067***	***3.62***	***82.07***	***95.52***
Harbour	DMEW	0.003	0.32	74.94	86.24	-0.003	0.28	74.80	86.69
	DT	0.171	-0.40	75.56	86.49	0.136	-0.83	74.56	86.35
	DMEW+DT	***0.175***	***-0.56***	***86.24***	***98.90***	***0.141***	***-1.01***	***85.00***	***98.83***
Mobile	DMEW	-0.022	0.85	74.95	86.20	-0.017	0.71	75.15	86.61
	DT	0.041	0.76	74.49	85.66	0.097	-0.63	74.20	85.94
	DMEW+DT	***0.047***	***0.57***	***86.00***	***99.01***	***0.106***	***-0.99***	***85.95***	***99.14***

Table 6.18. Comparison of the different proposals within this thesis - Main Profile and GOP = 8

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	0.022	0.62	72.27	89.35	0.007	0.48	68.38	85.58
	DT	0.668	-0.07	71.66	87.93	0.443	-1.09	71.20	88.23
	DMEW+DT	***0.666***	***-0.03***	***81.29***	***99.29***	***0.444***	***-1.12***	***79.34***	***99.16***
City	DMEW	-0.028	1.66	46.74	60.40	-0.120	3.62	49.16	61.81
	DT	0.063	1.81	71.12	87.99	0.018	0.31	71.29	88.31
	DMEW+DT	***0.065***	***1.75***	***79.89***	***99.11***	***0.020***	***0.23***	***79.31***	***99.20***
Foreman	DMEW	-0.010	0.92	41.83	51.83	-0.038	1.39	43.23	54.44
	DT	0.040	3.39	72.65	88.41	-0.171	3.21	71.47	88.42
	DMEW+DT	***0.040***	***3.39***	***79.40***	***99.71***	***-0.175***	***3.15***	***78.14***	***97.68***
Soccer	DMEW	-0.123	4.13	34.19	41.39	-0.111	5.74	30.60	38.76
	DT	-0.027	5.52	69.99	86.09	0.105	3.46	70.57	87.24
	DMEW+DT	***-0.016***	***4.97***	***75.72***	***99.43***	***0.094***	***4.10***	***76.38***	***95.55***
Harbour	DMEW	0.005	0.32	70.25	86.32	0.012	0.26	66.43	83.21
	DT	0.361	-0.46	70.30	86.57	0.244	-0.73	69.98	86.48
	DMEW+DT	***0.366***	***-0.66***	***80.29***	***98.90***	***0.248***	***-0.89***	***79.16***	***98.88***
Mobile	DMEW	-0.018	0.76	69.68	86.13	-0.018	0.77	66.21	82.84
	DT	0.022	0.68	70.39	86.31	0.212	-0.23	71.63	87.28
	DMEW+DT	***0.026***	***0.52***	***80.73***	***99.09***	***0.219***	***-0.53***	***81.15***	***99.22***

Table 6.19. Comparison of the different proposals within this thesis - Main Profile and GOP = 16

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	0.013	0.59	69.56	88.58	0.003	0.64	66.01	85.43
	DT	0.632	-0.06	68.34	86.72	0.337	-0.89	66.81	87.92
	DMEW+DT	***0.632***	***-0.06***	***76.12***	***97.80***	***0.337***	***-0.90***	***76.38***	***99.27***
City	DMEW	-0.108	2.73	50.16	64.01	-0.100	2.61	52.22	67.85
	DT	-0.009	1.92	68.91	87.18	-0.008	0.34	68.75	88.37
	DMEW+DT	***-0.007***	***1.87***	***75.42***	***97.62***	***-0.006***	***0.29***	***76.32***	***99.22***
Foreman	DMEW	-0.026	0.85	42.34	54.04	-0.035	1.30	44.99	58.56
	DT	0.096	2.72	67.62	86.39	-0.102	2.99	68.93	88.46
	DMEW+DT	***0.096***	***2.85***	***73.21***	***95.44***	***-0.106***	***3.23***	***75.14***	***97.62***
Soccer	DMEW	-0.087	4.61	31.04	39.81	-0.121	5.76	34.89	45.59
	DT	-0.029	5.06	67.15	85.06	0.117	3.66	68.13	87.36
	DMEW+DT	***-0.059***	***4.41***	***71.47***	***92.83***	***0.105***	***4.33***	***73.54***	***95.60***
Harbour	DMEW	0.023	0.33	67.03	85.45	0.011	0.31	64.40	83.43
	DT	0.172	-0.42	66.72	85.27	0.168	0.54	67.80	87.09
	DMEW+DT	***0.175***	***-0.52***	***75.18***	***97.41***	***0.171***	***-0.67***	***76.26***	***98.92***
Mobile	DMEW	-0.013	0.68	67.05	85.45	-0.016	0.79	66.48	86.45
	DT	0.033	0.53	67.29	85.28	0.185	0.10	68.39	87.22
	DMEW+DT	***0.038***	***0.33***	***75.33***	***97.58***	***0.192***	***-0.18***	***77.75***	***99.25***

Table 6.20. Comparison of the different proposals within this thesis - Main Profile and GOP = 32

		QCIF (15 Hz)				*CIF (30 Hz)*			
Sequence	**Proposal**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**		**ΔPSNR (dB)**	**ΔBitrate (%)**	**Time Saving (%)**	
				Full Seq.	***Partial***			***Full Seq.***	***Partial***
Hall	DMEW	-0.017	0.59	69.29	89.48	-0.002	0.75	64.79	85.26
	DT	0.626	0.29	70.31	88.83	0.166	-0.34	66.90	87.78
	DMEW+DT	***0.627***	***0.29***	***76.73***	***99.26***	***0.167***	***-0.34***	***74.88***	***98.57***
City	DMEW	-0.030	0.92	52.04	67.27	-0.119	5.70	47.41	62.73
	DT	-0.020	1.68	68.97	88.61	-0.011	0.46	67.60	87.78
	DMEW+DT	***-0.019***	***1.63***	***76.07***	***99.17***	***0.100***	***0.40***	***75.09***	***98.54***
Foreman	DMEW	-0.010	0.93	37.39	48.45	-0.034	1.22	40.78	54.09
	DT	-0.024	3.41	69.16	88.32	0.001	3.19	67.93	87.93
	DMEW+DT	***-0.024***	***3.50***	***74.42***	***96.70***	***-0.002***	***3.40***	***74.02***	***97.03***
Soccer	DMEW	-0.105	5.74	32.63	42.30	-0.142	5.96	29.09	38.87
	DT	0.018	5.79	66.12	86.12	0.089	3.62	67.17	86.92
	DMEW+DT	***-0.013***	***5.23***	***71.40***	***94.44***	***0.076***	***4.38***	***72.37***	***95.00***
Harbour	DMEW	0.035	0.39	66.57	86.07	0.016	0.40	62.90	82.83
	DT	0.371	0.16	68.13	87.39	0.229	-0.15	66.96	86.79
	DMEW+DT	***0.374***	***-0.02***	***76.30***	***98.96***	***0.231***	***-0.26***	***75.06***	***98.28***
Mobile	DMEW	-0.009	0.70	67.09	86.57	-0.015	0.82	62.61	82.45
	DT	0.102	1.03	67.08	86.74	0.144	0.46	66.63	86.59
	DMEW+DT	***0.107***	***0.86***	***75.71***	***99.16***	***0.150***	***0.19***	***75.75***	***98.58***

6.3.3 Comparison with state-of-the-art results

As mentioned in Section 3.3, different techniques have been proposed in the literature recently for transcoding from H.264/AVC to SVC focusing on temporal scalability in Baseline Profile. Our technique is capable of outperforming those solutions such as in [94][95]. In contrast to [94], 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 6.21. 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 ∆Bitrate, there is not numerical information in [94][95]. It should be noted that we have selected the same sequences used in references in order to make a fair comparison

Table 6.21. Comparison of other proposals for providing temporal scalability

Comparison with other proposals						
	GOP = 2 CIF - Foreman		***GOP = 8 QCIF***		***GOP = 8 CIF***	
Method	**∆PSNR (dB)**	**TS (%)**	**∆PSNR (dB)**	**TS (%)**	**∆PSNR (dB)**	**TS (%)**
Dziri et al. [94]	-0.500	47.00	--	--	--	--
Al-Muscati et al. [95]	-0.500	37.00	-0.200	55.20	--	62.10
DMEW proposal	-0.010	41.79	-0.027	51.90	-0.040	48.83
DT proposal	-0.059	58.12	0.086	67.04	0.084	68.36
Our combined proposal	***-0.058***	***65.21***	***0.089***	***78.44***	***0.089***	***79.03***

6.4 Conclusions

In this chapter, a combination of the techniques proposed during this thesis was presented. By reducing the complexity of ME and mode decision tasks in the encoder stage and combining these reductions, a 98% of time reduction is achieved where the proposal is applied while maintaining the coding efficiency. The joint proposal is valid for different profiles, GOP sizes and resolutions and even can achieved better results of bitrate and PSNR than the reference transcoder with the RDO option disabled.

CHAPTER 7

CONCLUSIONS, FUTURE WORK AND PUBLICATIONS

7.1 Conclusions

The objective of this thesis has been to propose a H.264/AVC-to-SVC transcoder with temporal scalability to provide bitstreams without this type of scalability with it. In order to achieve this objective, some specific technical contributions were presented. These techniques focus on the encoder stage of the transcoder because is the stage where most time is spent. An enumeration of the contributions and the main conclusions that have been obtained can be found in the following lines:

- At the beginning of this thesis, a study of the percentage of the encoding time spent to encode every temporal layer in different sequences was done. After that study, it was concluded that the two temporal layers with highest identifiers need around an 80% of the encoding time to be encoded, so the techniques developed during this thesis were applied to these temporal layers. Moreover, in each proposed technique, a study of the effect of how it affects the number of temporal layers where it is applied is presented.

- One of the tasks which take most of the time for re-encoding a video sequence is the ME process. The first approach presented in this thesis to accelerate the encoding stage of the transcoder focuses in this task. The proposal reuses information collected from the decoder stage such as MVs to build a new reduced search area that adjusts dynamically depending on the temporal layer.

  - First of all, a circumference as the reduced search area was presented. This circumference was build using the information of the movement extracted from the decoding stage. In particular, the average of the length of the MVs of each MB was used to calculate the radius of that circumference.
  - After building an initial reduced search area, some adjusts needed to be made because of the different of the prediction structures in H.264/AVC without temporal scalability and SVC. This initial search area was adjusted by multiplying the radius by a factor to be adjusted depending on the temporal layer which the approach is applied.
  - This proposed mechanism was evaluated using H.264/AVC videos encoding with different profiles, different resolutions, and different GOP sizes and applying the proposal to different combinations of temporal layers. The results show that the proposed dynamic ME window is able to maintain the same quality while considerably reducing the ME computational complexity by as much as 70%. The reduction in computational cost has no impact on the quality and bitrate of the transcoded video. Moreover, the results show that the best combination of temporal layers for applying the proposal was the two temporal enhancement layers with a highest identifier. Regarding the GOP size used, this size has a negligible impact in the results obtained.

- The other part of the interprediction process in the encoding stage which spends a major computational time is the mode decision task. The proposal uses ML techniques to exploit the correlation between some information of H.264/AVC such as residual, coding modes, MVs, etc. and the mode decision of SVC.
  - First of all, a decision tree using WEKA tool was build. For that, as said previously, some information from decoding and encoding stage was collected and a decision tree was made to decide which MB partitions will be checked by the encoding stage depending on some thresholds. These thresholds were adapted based on the quantization parameter selected in SVC encoding stage, so the decision tree is unique for every profile and only needs to be adjusted for every QP. This decision tree narrowed down the number of MB partitions to be checked.
  - Such as in the dynamic ME window, this idea was evaluated H.264/AVC videos encoding with different profiles and with different resolutions and applied to various GOP sizes and different combinations of temporal layers. The results show that the proposed algorithm is able to maintain a good picture quality while considerably reducing the computational complexity by as much as 84%. The reduction in computational cost has negligible impact on the quality of the transcoded video. Moreover, as in the other proposal, the results show that the best combination of temporal

layers for applying the technique was the two temporal enhancement layers with a highest identifier. Regarding the GOP size used, this size has a negligible impact in the results obtained.

- All the techniques proposed during this thesis are combined to create an *H.264/AVC-to-SVC Transcoder.* Simulations were performed on six standard video sequences in different resolutions, GOP sizes from 2 to 32 and Baseline and Main Profiles. The transcoder can reduce extremely computational cost up to 99% on average in the temporal layers where the technique is applied with a negligible PSNR degradation and slight increase in bitrate. As in the previous proposals, the GOP size variation does not affect significantly to the results.

- A comparative study with other existing methods for H.264/AVC-to-SVC transcoding focused in temporal scalability presented in the literature has been carried out. The results have shown that the proposed approach achieves the best results.

7.2 Future Work

During the development of this thesis, a transcoder from H.264/AVC-to-SVC with temporal scalability has been proposed. This proposal is composed by two big proposals that then were adjusted to work together. At this point, different options can be explored in order to improve the proposed H.264/AVC-to-SVC transcoder or to extend the transcoding from other standards to SVC.

- *Developing a technique for accelerating the Intra prediction.* In general, intra-frame prediction is very complex process. As in the encoding stage of the proposed transcoder, this type of prediction is done, another way to accelerate the transcoding process is developing a technique to reduce the time used by intra-prediction.

- *Extending the transcoder to introduce spatial and quality scalability*. At this point, the transcoder proposed provides temporal scalability, although SVC can further provide spatial and quality scalability. As the information of the layers depends on the information of the others, ML techniques could be used for reduce the time necessary for encoding spatial and/or quality layers.

- *Simulating the transmission of the transcoded bitstream over a network.* As the resulting bitstream is a scalable bitstream (in this case with different frame rates as is with temporal scalability) simulations using Opnet, Omnet or other simulation software could be run to see the behaviour of the scalable bitstream when is transmitted over a network with devices with different characteristics.

- *Parallelizing part of the transcoder.* Another way to accelerate the transcoder is to reduce the encoding stage by parallelizing some tasks and running using GPUs or Multicores.

- *Hardware implementation of the transcoder.* A hardware implementation for evaluation the proposed transcoder in a real scenario could be developed using programmable electronic devices such as CPLD, PLDs or FPGAs.

- *Adapting the proposed transcoder for working with RDO and with different methods of ME.* The proposed transcoder has been designed for SAE cost (RDO disabled) and with Full Search in ME task. The transcoder could be adapted for working with RDO enabled and other methods of ME.

- *Adapting the proposals to make them work for transcoding from other standards.* As nowadays exist a large variety of devices with different characteristics and the video contents are already encoded in different standards, another research line could be generalize the techniques proposed for accelerating ME and mode decision tasks to be used for transcoding from other existing standards, even with future standards such as H.265.

7.3 Publications

The different proposals and results included in this thesis have lead to the publication of different journal articles and the participation on international and national conferences. These contributions are listed in the following sections accompanied with a brief description of each one.

7.3.1 Journals Indexed in Journal Citation Reports

1. 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," In: IEEE Transactions on Consumer Electronics, vol.57, no.1, pp. 239-246, February 2011. Impact: 1.057 (JCR 2010). Occupy the 30/80 on Telecommunication category.

This paper proposes a technique for transcoding from H.264/AVC-to-SVC providing temporal scalability. Focusing on the ME task, the proposed transcoder reuses the motion information from the H.264/AVC decoder and re-uses it to reduce the search area of the SVC encoder. Therefore the SVC encoding algorithm can be further reduced. This technique is developed for Baseline Profile. This journal paper summarizes the main results of Section 4.2 regarding Baseline Profile.

2. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, and A. Garrido. "Video Transcoding for Mobile Digital Television," In: Telecommunication Systems, Springer. Accepted. Impact: 0.670 (JCR 2010). Occupy the 46/80 on Telecommunication category.

> In this paper an adaptation for using in Main Profile the previous work of the IEEE Transaction on Consumer Electronics is presented. This journal paper summarizes the results of Section 4.2. regarding Main Profile.

3. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido, and R.Van de Walle. "On the Impact of the GOP Size in a Temporal H.264/AVC-to-SVC Transcoder in Baseline and Main Profile," In: Multimedia Systems, Springer. Accepted. Impact: 1.176 (JCR 2010). Occupy the 38/97 on Computer Science, Theory & Methods category.

> In this paper, a complete study of the proposal of the Dynamic ME Window approach is done. Both, Baseline and Main Profile are tested and the GOP size was varied for seeing its impact in the proposal. All the results of Chapter 4 are included.

4. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, and A. Garrido. "Scalable Video Transcoding for Mobile Communications," In: Telecommunication Systems, Springer. Accepted. Impact: 0.670 (JCR 2010). Occupy the 46/80 on Telecommunication category.

> This paper evaluates jointly the both approaches developed in the proposed transcoder in Baseline Profile: the dynamic ME window and the fast MB mode decision algorithm to reduce the interprediction. Both approaches are evaluated in Section 6.2.1.

7.3.2 International Conference Proceedings

1. R. Garrido-Cantos, J. L. Martinez, P. Cuenca, and A. Garrido. "An Approach for an AVC to SVC Transcoder with Temporal Scalability" In: Lecture Notes in Computer Science (LNCS), Proceedings of the 5th International Conference on Hybrid Artificial Intelligence Systems (HAIS) 2010, Vol. 6077-2 pp. 225–232. San Sebastián, Spain, June 2010.

> This paper presents the initial approach for the reduced ME search window. Then, some adjusts were done in the proposal to make the final technique presented in this thesis.

2. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido, and R. Van de Walle. "Video Adaptation for Mobile Digital Television." In: 3rd Joint

IFIP Wireless and Mobile Networking Conference (WMNC 2010), Budapest, Hungary, October 2010.

This paper is the previous work of the Telecommunication Systems Journal publication depicted above. This paper was selected in WMNC 2010 conference to be extending and submitting it to the Telecomunication Systems Journal. This journal paper summarizes the main results of section 4.2. regarding Baseline Profile.

3. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido, and R. Van de Walle. "On the impact of the GOP size in an H.264/AVC-to-SVC transcoder with temporal scalability." In: 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2010), Paris, France, November 2010.

This paper is the previous work of the Multimedia Systems Journal publication depicted above. This paper was selected by the organizers of the MoMM 2010 conference to be extending. This journal paper summarizes the main results of the Chapter 4 focusing on the impact of the GOP size.

4. R. Garrido-Cantos, J. De Cock, J.L. Martinez, S.Van Leuven, P. Cuenca, A. Garrido, and R. Van de Walle. "An H.264/AVC to SVC Temporal Transcoder in baseline profile". In: 2011 IEEE International Conference on Consumer Electronics (ICCE 2011), Las Vegas, NV, USA, January 2011.

This paper is an initial version of the proposal for reducing the search window area in Baseline Profile. In this version, only GOP size of 2 was tested with a reduced number of sequences.

5. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido, and R. Van de Walle. "Video Low Complexity Adaptation for Mobile Video Environments using Data Mining." In: 4rd Joint IFIP Wireless and Mobile Networking Conference (WMNC 2011), Toulusse, France, October 2011. ***Best Paper Award****.*

This article proposes an algorithm to determine a sub-set of MB mode coded partition for the SVC encoding as part of the proposed H.264/AVC-to-SVC transcoder. Based on the correlation of previous data calculated in the H.264/AVC decoder stage, the proposed algorithm fits to a sub-set of possible partitions reducing, therefore, the complexity. This proposal was applied in Baseline Profile. These results are shown in Section 5.2.2. This paper is the previous work of the Telecommunication Systems Journal publication depicted above. This paper was selected best paper award of the WMNC 2011 conference.

The editors proposed extending this work and submitting it to the Telecomunication Systems Journal.

6. R. Garrido-Cantos, J. De Cock, S. Van Leuven, P. Cuenca, A. Garrido, and R. Van de Walle.“Fast Mode Decision Algorithm for H.264/AVC-to-SVC Transcoding with Temporal Scalability.” In: 18th International Conference on MultiMedia Modeling (MMM2012), Klagenfurt (Austria), January 2012.

This paper presents an adaptation of the proposal developed for Baseline profile in the previous publication to be used with different GOP sizes in Baseline Profile. These results are shown in Section 5.2.2.

7.3.3 National Conferences Proceedings with peer review

1. R. Garrido-Cantos, J. L. Martinez, P. Cuenca, and A. Garrido. “On the Impact of Temporal Layers in an Improved AVC to SVC Transcoder” In: I Workshop on Multimedia Data Coding and Tranmission (WMDCT 2010), Valencia, Spain, September 2010.

This work is part of the proposal based on the ME computing complexity reduction. This paper evaluates the performance of the transcoder in Main Profile depending on the number of temporal layers where the proposal is applied. The results are included in 4.4.

7.3.4 Under Review

1. R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, and A. Garrido. ”H.264/AVC-to-SVC Temporal Transcoding using Machine Learning,” submitted to the 16th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES2012).

This paper presents the proposal of using a decision tree build with WEKA for narrowing down the decision modes checked by the encoder stage of the transcoder. The part presented in this paper focus on this technique used in Main Profile. The results are included in Section 5.2.3.

7.3.5 Other Publications

Journals Indexed in Journal Citation Reports

1. S. Van Leuven, J. De Cock, R. Garrido-Cantos, J.L. Martínez, and R. Van de Walle. “Generic Techniques to Reduce SVC Enhancement Layer Encoding Complexity,” In: IEEE Transactions on Consumer Electronics, vol.57, no.2, May 2011.

International Conference Proceedings

1. 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," In: 6th International ICST Mobile Multimedia Communications Conference (MobiMedia 2010), Lisbon, Portugal, September 2010. ***Best Paper Award**.*

2. S. Van Leuven, G. Van Wallendael, J. De Cock, R. Garrido-Cantos, J.L. Martinez, P. Cuenca, and R. Van de Walle. "Generic techniques to improve SVC enhancement layer encoding," In: 2011 IEEE International Conference on Consumer Electronics (ICCE 2011), Las Vegas, NV, USA, January 2011.

3. S. Van Leuven, G. Van Wallendael, J. De Cock, R. Van de Walle, R. Garrido-Cantos, J.L. Martinez, P. Cuenca. "A Low-Complexity Closed-Loop H.264/AVC to Quality-Scalable SVC Transcoder." In: 2011 IEEE International Conference on Digital Signal Processing (DSP 2011), Corfu, Greece, July 2011.

4. S. Van Leuven, G. Van Wallendael, J. De Cock, R. Van de Walle, R. Garrido-Cantos, J.L. Martinez, P. Cuenca. "Combining Open- and Closed-Loop Architectures for H.264/AVC-to-SVC Transcoding." In: 2011 IEEE International Conference on Image Processing (ICIP 2011), Brussels, Belgium, September 2011.

APPENDIX I

GLOSSARY

AIR	Adaptive Intra Refresh
ARFF	Attribute-Relation File Format
ASO	Arbitrary Slice Order
AVC	Advanced Video Coding
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CD-ROM	Compact Disk - Read Only Memory
CGS	Coarse Grain Scalability
CIF	Common Intermediate Format
CPDT	Cascade Pixel Domain video Transcoder
CPLD	Complex Programmable Logic Device
DCT	Discrete Cosine Transform
DMEW	Dynamic Motion Estimation Window
DP	Data Partitioning
DT	Decision Tree
DVC	Distributed Video Coding
DVD	Digital Versatile Disk
FCS	Feedback Control Signalling
FGS	Fine Grain Scalability
FMO	Flexible Macroblock Order
FPGA	Field Programmable Gate Array
FRExt	Fidelity Range Extension
GOP	Group of Pictures
GPRS	General Packet Radio Services
GPU	Graphics Processing Unit

GUI	Graphical User Interface
H.261	A video coding standard
H.263	A video coding standard
H.263+	A video coding standard
H.263++	A video coding standard
H.264/AVC	A video coding standard
H.265	A video coding standard
HT	Hadamard Transform
IDR	Instantaneous Decoding Refresh
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JCR	Journal Citation Report
JM	Joint Model
JSVM	Joint Scalable Video Model
JVT	Joint Video Team
LNCS	Lecture Notes in Computer Science
MB	Macroblock
MB-AFF	Macroblock Adaptive Frame/Field
MC	Motion Compensation
ME	Motion Estimation
MGS	Medium Grain Scalability
ML	Machine Learning
MPEG	Motion Pictures Expert Group
MPEG-1	Multimedia Coding Standard
MPEG-2	Multimedia Coding Standard
MPEG-4	Multimedia Coding Standard
MV	Motion Vector
NAL	Network Abstraction Layer
PAFF	Picture Adaptive Frame/Field
PLD	Programmable Logic Device
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantification Parameter
QQVGA	Quarter-QVGA
QVGA	Quarter VGA
RD	Rate Distortion
RD	Redundant Picture
RDO	Rate Distortion Optimization
RP	Redundant Pictures
RPS	Redundant Picture Selection

RPI	Redundant Picture Information
SI	Switching Intra
SNR	Signal to Noise Ratio
SP	Switching Predictive
SVC	Scalable Video Coding
TL	Temporal Layer
TV	Television
VCEG	Video Coding Expert Group
VCL	Video Coding Layer
VGA	Video Graphics Array
VHS	Video Home System
VLC	Variable Length Coding

BIBLIOGRAPHY

[1] Advanced Television System Committee: ATSC-Mobile DTV Standard, A/153 ATSC Mobile Digital Television System. October 2009.

[2] Digital Video Broadcasting *(DVB)*. Transmission System for Handheld Terminals (DVB-H), ETSI EN 302 304 V1.3.1. 2009.

[3] 3GPP: TS 23.246 V9.4: Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description. March 2010.

[4] ISO/IEC SC29 WG11 13818. Generic Coding of Moving Pictures and Associated Audio. 1994.

[5] ISO/IE-C 14496-2, Coding of audio-visual objects – Part 2: Visual, 2001.

[6] DivX: http://www.divx.com/

[7] XviD: http://www.xvid.org/

[8] ITU-T and ISO/IEC JTC 1: Advanced Video Coding for Generic Audiovisual Services. ITU-T Rec. H.264/AVC and ISO/IEC 14496. 2003.

[9] Blackberry smartphone: http://es.blackberry.com//

[10] Apple iPhone. http://www.apple.com/es/iphone/

[11] Motorola Xoom: http://www.motorola.com/

[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] Digital Video Broadcasting (DVB). IP Datacast over DVB-H: Content Delivery Protocols (CDP) - A101r1 (dTS 102 472 v1.3.1). 2009.

[14] Advanced Television System Committee: ATSC-Mobile DTV Standard, Part 7 – AVC and SVC Video System Characteristics. October 2009.

[15] ITU-T Recommendation. H.120 Video, Video CODEC for audiovisual services, 1990.

[16] ITU-T Recommendation. H.261 Video, Video CODEC for audiovisual services at px64 kbits/s. 1993.

[17] ISO/IEC JTC1 CD 11172 “Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5 Mbits/s”. Part 1: Systems; Part 2: Video; Part 3: Audio; Part 4: Conformance Testing ISO, 1993.

[18] ITU-T Recommendation H.263, Video Coding for low bit rate communication, version 2, 1998.
[19] VCEG, Draft requirements for "EPVC" enhanced performance video coding project
[20] 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), January 2012.
[21] R. Schäfer, T. Wiegand and H. Shwarz, "The emerging H.264/AVC standard", EBU Technical Review, 2003.
[22] A. Bovik, "The Essencial Guide to Video Transcoding", 2nd Edition, Elsevier, USA, 2009.
[23] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Extension of the H.264/MPEG-4 AVC Video Coding Standard". Joint Video Team, Doc. JVT-W132, San Jose, CA, USA, 2007.
[24] H. Schwarz, D. Marpe and T. Wiegand, "Analysis of Hierarchical B pictures and MCTF," IEEE Int. Conf. ICME and Expo, Toronto, 2006.
[25] A. Vetro, C. Christopoulos and H. Sun, "Video Transcoding Architectures and Techniques: an Overview," IEEE Signal Processing Magazine. 18--29, 2003.
[26] I. Ahmad, X. Wei, Y. Sun, and Y. Zhang, "Video Transcoding: An Overview of Various Techniques and Research Issues," IEEE Transactions on Multimedia, vol. 7, no. 5, 2005.
[27] J. Xin, C. Lin, and M. Sun, "Digital Video Transcoding," Proceedings of the IEEE, 2005.
[28] N. Bjork and C. Chisopoulos, "Transcoder Architectures for Video Coding," IEEE Transactions on Consumer Electronics, vol. 44, no. 1, pp. 88-98, February 1998.
[29] S. Moiron, M. Ghanbari, P. Assunção, and S. Faria, "Video Transcoding Techniques," Recent Advances in Multimedia Signal Processing and Communications, 2009.
[30] H. Sun, W. Kwok and J. W. Zdepski, "Architecture for MPEG compressed video down-scaling," IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 2, pp. 191-199, April 1996.
[31] P.A.A. Assunçäo and M. Ghanbari, "Transcoding of Single-layer MPEG Video into Lower Rates," IEEE Proceedings Image Signal Process, vol. 144, no. 6, December 1997.
[32] O. Werner, "Requantisation for Transcoding of MPEG-2 intraframes," IEEE Transactions on Image Processing, vol. 8, no. 2, pp 179-191, February 1999.
[33] D. Lefol, D. Bull, N. Canagarajah, "Performance Evaluation of Transcoding Algorithms for H.264," IEEE Transactions on Consumer Electronics, vol. 52 no. 1, 2006.
[34] B. Shen, "Perfect Requantization for Video Transcoding," Multimedia Tools and Applications, vol. 35, no. 2, pp 163–173, 2007.
[35] J. De Cock, S. Notebaert, P. Lambert, D. De Schrijver, R. Van de Walle, "Requantization Transcoding in Pixel and Frequency Domain for intra 16×16 in

H.264/AVC," Lecture Notes in Computer Science, Advanced Concepts for Intelligent Vision Systems, ACIVS, 2006.

[36] J. De Cock, S. Notebaert, R. Van de Walle, "Combined SNR and Temporal Scalability for H.264/AVC using Requantization Transcoding and Hierarchical B Pictures," IEEE International Conference on Multimedia and Expo (ICME), pp 448–451, 2007.

[37] S. Notebaert, J. De Cock, S. Beheydt, J. De Lameillieure, and R. Van de Walle, "Mixed Architectures for H.264/AVC Digital Video Transrating," Multimedia Tools and Applications, vol. 44, no. 1, pp 34-69, 2009.

[38] C.-W. Lin, T.-J. Liou y Y.-C. Chen, "Dynamic Rate Control in Multipoint Video Transcoding," IEEE International Symposium on Circuits and Systems, Geneva, Switzerland, May 2000.

[39] C. Deknudt, P. Corlay; A.-S. Bacquet, M. Zwingelstein-Colin; F.-X. Coudoux, "Reduced Complexity H.264/AVC Transrating based on Frequency Selectivity for High-definition Streams," IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp 2430 - 2437, 2011.

[40] K-H. Tan and Ganbari, "Layered Image Coding using the DCT pyramid," IEEE Transactions on Image Process, vol. 4, no. 4, 1995.

[41] R. Mohan, J.R Smith, and C. Li, "Adapting Multimedia Internet Content for Universal Access," IEEE Transactions on Multimedia, vol.1, no. 1, 1999.

[42] B. Shen, I.K. Sethi and B. Vasude, "Adaptative Motion-vector Resampling for Compressed Video Down-scaling," IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 6, pp. 929-936, September 1999.

[43] P. Yin, M. Wu and B. Liu, "Video Transcoding by Reducing Spatial Resolution," Proceedings of the International Conference on Image Processing (ICIP´00), Vancouver, BC, Canada, September 10-13 2000.

[44] T. Shanableh and M. Ghanbari, "Heterogeneous Video Transcoding to Lower Spatial-temporal Resolutions and different Encoding Formats," IEEE Transactions on Multimedia, vol. 2, no. 2, 2000.

[45] B. Shen, S. Roy; "A very Fast Video Spatial Resolution Reduction Transcoder," 2002 IEEE International Conference on Acoustics, Speech and Signal Processing, pp 1989-1992, Orland, USA, 2002.

[46] H.-S. Kong, A. Vetro, H. Sun, "Combined Rate Control and Mode Decision Optimization for MPEG-2 Transcoding with Spatial Resolution Reduction," Proceedings of the 2003 International Conference on Image Processing, vol. 1, pp 164, Barcelona, Spain, 2008.

[47] A. Yusuf, M. Murshed, L.S. Dooley, "An HVS-based MV Composition Algorithm for Spatial Resolution Transcoding," Proceedings of 2004 International Conference on Information Technology, Coding and Computing, pp 682 - 688, Las Vegas, USA, 2008.

[48] J. De Cock, S. Notebaert; K. Vermeirsch, P. Lambert; and R. Van de Walle, "Efficient Spatial Resolution Reduction Transcoding for H.264/AVC," 15th IEEE

International Conference on Image Processing, pp 1208-1211, San Diego, California, 2008.

[49] G. Hongxing ; W. Jie ; T. Ting ; W. Hengwei ; Y. Shengsheng;"Transrating-assisted Spatial Resolution Reducing for MPEG-4 Video Streams," Proceedings of the 10th International Conference of Signal Procesing, pp 1149 - 1152, Beijing, China, 2010.

[50] A.-S. Bacquet; C. Deknudt, P. Corlay, F.- X.Coudoux, M. Gazalet, "Low Complexity H.264/AVC Spatial Resolution Transcoding in the Transform Domain," Proceedings of the 17th IEEE International Conference on Electronics Circuits and Systems, pp 491 - 494, Athens, Greece, 2010.

[51] J.-N. Hwang and T.-D. Wu, "MV Re-estimation and Dynamic Frame-skipping for Video Transcoding," Proceedings of the 32nd Conference on Signals, Systems and Computers, vol. 2, pp. 1606-1610, Pacific Grove, USA, November 1998.

[52] J. Youn, M.-T. Sun and C.-W. Lin, "Motion Vector Refinement for High-Performance Transcoding," IEEE Transactions on Multimedia, vol. 1 no. 1, pp. 30-40. March 1999.

[53] K.-T. Fung, Y.-L. Chang y W.-C. Siu, "Dynamic Frame Skipping for High-Performance Transcoding". Proceedings of the International Conference on Image Processing, ICIP´01, Thessalonica, Greece, October 2001.

[54] M.-J. Chen, M.-C. Chu, and C.-W. Pan, "Efficient Motion-estimation Algorithm for Reduced Frame-rate Video Transcoder," IEEE Transactions on Circuits Systems and Video Technology, vol. 2, no. 4, 2002.

[55] Y.-K. Lee; S.-S. Lee; Y.-L. Lee, "MPEG-4 to H.264 Transcoding using MB Statistics," 2006 IEEE International Conference on Multimedia Expo, pp 57-60, Toronto, Canada, 2006.

[56] H. Shu; L.-P. Chau, "Variable Frame Rate Transcoding considering Motion Information," 2005 IEEE International Symposium on Circuits Systems pp 2144-2147, Kobe, Japan, 2005.

[57] G.-R. Kwon; S.-H. Park; J.-W. Kim; S.-J. Ko, "Real-Time R-D Optimized Frame-Skipping Transcoder for Low Bit Rate Video Transmission," The 6th IEEE International Conference on Computer and Information Technology, pp 177 - 185, Seoul, Corea, 2006.

[58] Y-K. Lee, S.-S. Lee and Y.-L. Lee, "MPEG-4 to H.264 Transcoding with Frame Rate Reduction," Multimedia Tools and Applications, vol. 35, no. 2, pp. 147-162, 2007.

[59] C. T. Hsu, C. H. Yeh, and M. J. Chen, "Effective Frame Rate Decision by Lagrange Optimization for Frame Skipping Video Transcoding," Lecture Notes in Computer Science, vol. 5359, pp. 551-560, 2008.

[60] C. T. Hsu, C. H. Yeh, C. Y. Chen, and M. J. Chen, "Arbitrary Frame Rate Transcoding through Temporal and Spatial Complexity," IEEE Transactions on Broadcasting, vol. 55, no. 4, pp. 767-775, Dec. 2009.

[61] A. Sadka, "Compressed Video Communications", Wiley & Sons, 2002.

[62] R. Swann and N.G. Kingsbury, “Transcoding of MPEG-II for Enhanced Resilience to Transmission Errors,” Proceedings of the IEEE International Conference on Image Processing, ICIP´96, pp. 813-816, Lausanne, Switzerland, September 1996.

[63] G. de los Reyes, A.R. Reibman, S.F. Chang and J.C.I. Chiang, “Video Transcoding for Resilence in Wireless Chanels,” Proceedings of the IEE International Conference on Image Processing, ICIP´98, pp. 338-342, Chicago, USA, October 1998.

[64] A. Dogan, A. Cellatoglu, A.H. Sadka and A.M. Kondoz, “Error-resilent MPEG-4 Video Transcoder for Bit Rate Regulation,” Proceedings of the Fifth World Multi-conference on Systemics, Cybernetics and Informatics, SCI´2001, vol. XII, pp. 312-317 Orlando, USA, July 2001.

[65] M. Xia, A. Vetro, B. Liu, and H. Sun, “Rate-distortion Optimized Bit Allocation for Error Resilient Video Transcoding,” IEEE International Symposium on Circuits and Systems, vol. 3, 2004.

[66] W.-N. Lie; H.-C. Yeh; Z.-W. Gao; P.-C. Jui; “Error-Resilience Transcoding of H.264/AVC Compressed Videos,” 2007 IEEE International Symposium on Circuits and Systems, pp 1509 - 1512, New Orleans, USA, 2007.

[67] T. Shanableh, T. May, F. Ishtiaq, “Error Resiliency Transcoding and Decoding Solutions using Distributed Video Coding Techniques,” Signal Processing: Image Communication, vol. 23, no. 8, pp 610-623, 2008.

[68] Y.-L. Chan ; H.-K. Cheung ; W.-C. Siu, “Compressed-Domain Techniques for Error-Resilient Video Transcoding Using RPS,” IEEE Transactions on Image Processing, vol. 18, no.2, pp 357 - 370, 2009.

[69] D. Zhang ; Y. Guo ; H. Li ; C. Wen Chen, “Error Resilient Scalability for Video Bit-stream over Heterogeneous Packet Loss Networks,” 2010 IEEE International Symposium on Circuits and Systems, pp 4201 - 4204, Paris, France, 2010.

[70] Y. Zhou ; “An Adaptive Error-resilient Scheme for Temporal Video Transcoding”. The 2nd International Conference on Multimedia Technology, pp 141 - 144, Hangzhou, China, 2011.

[71] S. Dogan, A.H. Sadka and A.M. Kondoz, “Tandeming/transcoding issues between MPEG-4 and H.263,” Mobile and Personal Satellite Communications 3, Proceedings of the Third European Workshop on Mobile/Personal Satcoms, EMPS´98, pp. 339-346, Springer-Verlag, U.K 1999.

[72] N. Feamster and S. Wee, “An MPEG-2 to H.263 Transcoder,” Proceedings of the Second Multimedia Systems and Applications, SPIE-International Society for Optical Engineering, 3845, 164-175, Boston, USA, September 1999.

[73] J.Bialkowski, A. Kaup and K. Illgner, “Fast Transcoding of Intra Frames between H.263 and H.264,” Proceedings of IEEE International Conference on Image Processing, ICIP´04, pp. 2785-2788, Singapore 2004.

[74] E. Peixoto, R. L. de Queiroz, and D. Mukherjee, “A Wyner-Ziw Video Transcoder,” IEEE Transactions of Circuits and Systems for Video Technology, vol. 20, no. 2, pp 189 - 200, 2008.

[75] G. Fernandez- Escribano, H. Kalva, P. Cuenca, L. Orozco-Barbosa, and A. Garrido, “A Fast MB Mode Decision Algorithm for MPEG-2 to H.264 P-Frame Transcoding,” IEEE Transactions on Circuits and Systems for Video Technology, vol.18, no. 2, pp. 172-185, 2008.

[76] G. Fernández-Escribano, J. Bialkowski, J.A. Gámez, H. Kalva, P. Cuenca, L. Orozco-Barbosa, A. Kaup,: “Low-Complexity Heterogeneous Video Transcoding Using Data Mining,” IEEE Transactions on Multimedia, vol. 10, no. 2, pp. 286- 299, 2008.

[77] J. L Martinez, G. Fernandez-Escribano., H. Kalva, W.A.C Fernando, and P. Cuenca, “Wyner-Ziv to H.264 Video Transcoder for Low Cost Video Communications,” IEEE Transaction on Consumer Electronics, vol. 55, no. 3, pp 1453 - 1461, 2009.

[78] Q. Tang; P. Nasiopoulos, “Efficient Motion Re-Estimation with Rate-Distortion Optimization for MPEG-2 to H.264/AVC Transcoding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, pp 262 - 274, 2010.

[79] J. Xiaocong; J. Sun; J. Zhou ; and Z. Wang, “Fast Video Transcoding from AVS to H.264/AVC with Reduced Resolution,” IEEE International Conference on Signal Processing Communications and Computing, pp 1- 6, Xi'an, China, 2011.

[80] A. Corrales-García, J. L. Martínez, G. Fernández-Escribano and F. J. Quiles, “Scalable Mobile-to-Mobile Video Communications Based on an Improved WZ-to-SVC Transcoder,” 18th International Conference on MultiMedia Modeling (MMM2012), Klagenfurt, Austria, 2012.

[81] N. Yeadon, N. Davies, A. Friday, and G. Blair, “Supporting Video in Heterogeneous Mobile Environments,” Proceedings of the ACM Symposium on Applied Computing, SAC´98, Marriott Marquis, Atlanta, USA, February 1998

[82] G. Iannaccone, “Reducing Replication of Data in a Layered Video Transcoder,” Proceedings of the First International Networked Group Communication COST´264 Workshop, NGC´99, pp. 144-151, Pisa, Italy, November 1999.

[83] S. Dogan, A.H. Sadka, and A.M. Kondoz, “MPEG-4 Video Transcoder for Mobile Multimedia Traffic Planning,” Proceedings of the IEEE Second International Conference on 3G Mobile Communication Technologies, 3G 2001, pp. 109-113, London, UK, March 2001.

[84] I.M Martin, “Hybrid Transcoding for Adaptive Transmission of 3D Content,” 2002 IEEE International Conference on Multimedia and Expo pp 373 - 37, Lausanne, Switzerland, 2002.

[85] I. Shin; and K. Koh, “Hybrid Transcoding for QoS Adaptive Video-on-demand Services,” IEEE Transactions on Consumer Electronics, vol.50, no.2, pp 732 - 736, 2004.

[86] 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.

[87] J. De Cock, S. Notebaert, P. Lambert and R. Van de Walle, "Bridging the Gap: Transcoding from Single-Layer H.264/AVC to Scalable SVC Video Streams," ACIVS 2007, LNCS 4678, pp. 652–662, 2007.

[88] J. De Cock, S. Notebaert, P. Lambert and R. Van de Walle, "Advanced Bitstream Rewriting from H.264/AVC to SVC," ICIP 2008.

[89] 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 no. 7, pp.1209--1224, 2009.

[90] 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.

[91] 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, Corfu, Greece, 2011.

[92] 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, Brussels, Belgium, 2011.

[93] R. Sachdeva, S. Johar and E. Piccinelli, "Adding SVC Spatial Scalability to Existing H.264/AVC Video," 8th IEEE/ACIS International Conference on Computer and Information Science, Shanghai, 2009.

[94] 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.

[95] H. Al-Muscati, and F. Labeau, "Temporal Transcoding of H.264/AVC Video to the Scalable Format," 2nd International Conference on Image Processing Theory Tools and Applications, Paris, France, 2010.

[96] A. Segall, "SVC-to-AVC Bit-Stream Rewriting for Coarse Grain Scalability," Klagenfurt, Austria, Joint Video Team, Doc. JVT-T061, Jul. 2006.

[97] A. Segall and J. Zhao, "Bitstream Rewriting for SVC-to-AVC Conversion," ICIP 2008.

[98] P. Amon, L. Haoyu, A. Hutter, D. Renzi, S. Battista, "Scalable Video Coding and Transcoding,"2008. IEEE International Conference on Automation, Quality and Testing, Robotics, 2008.

[99] 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, no. 1, February 2011.

[100] 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. In press.

[101] R. Garrido-Cantos, J. L. Martínez, P. Cuenca, and A. Garrido, "On the Impact of Temporal Layers in an Improved AVC to SVC Transcoder", I Workshop on Multimedia Data Coding and Transmission, Valencia, Spain, 2010.

[102] R. Garrido-Cantos, J. De Cock, J.L. Martínez, S. Van Leuven, P. Cuenca, A. Garrido, and R. Van de Walle. On the Impact of the GOP Size in an H.264/AVC-to-SVC Transcoder with Temporal Scalability. 8th International Conference on Advances in Mobile Computing & Multimedia, Paris, France, 2010.

[103] Joint Model (JM) Reference Software: http://iphome.hhi.de/suehring/tml/

[104] Joint Scalable Video Model (JSVM) Reference Software: http://ip.hhi.de/imagecom_G1/savce/index.htm

[105] 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.

[106] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques," 3rd Edition, Morgan Kaufmann, San Francisco, 2011.

[107] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update; SIGKDD Explorations," vol. 11, no. 1, 2009.

[108] 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), Toulouse, France, 2011.

[109] R. Garrido-Cantos, J. De Cock, S. Van Leuven, P. Cuenca, A. Garrido and R. Van de Walle, "Fast Mode Decision Algorithm for H.264/AVC-to-SVC Transcoding with Temporal Scalability," 18th International Conference on MultiMedia Modeling (MMM2012), Klagenfurt, Austria, 2012.

[110] W. W. Cohen, "Fast Effective Rule Induction". 20th International Conference on Machine Learning, pp. 115-123, 1995.

[111] G. Fernandez-Escribano, "Low Complexity MPEG-2 to H.264 Transcoding", PhD. Thesis, University of Castilla-La Mancha, Albacete, Spain, 2007.

[112] R. Garrido-Cantos, J. De Cock, J. L. Martínez, S. Van Leuven, P. Cuenca, and A.Garrido "Scalable Video Transcoding for Mobile Communications," Telecommunication Systems. In press.