UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Department of Signal Theory and Communications

Ph.D. Dissertation

Optimization of Segmentation Based Video Sequence Coding Techniques: Application to Content Based Functionalities

Author: Josep Ramon Morros i Rubió
Advisor: Prof. Ferran Marqués

Barcelona, October 2004

A l'Alicia

Me dejar colmar
por las cosas menudas
- no quiero ya la llave de las inmensidades -
para alcanzar mi cielo
y para que mi infierno sea mío
transparente y sabroso.

Alicia Castillo

Resum

En aquest treball s'estudia el problema de la compressió de video utilitzant funcionalitats basades en el contingut en el marc teòric dels sistemes de codificació de seqüències de video basats en regions. Es tracten bàsicament dos problemes: El primer està relacionat amb com es pot aconseguir una codificació òptima en sistemes de codificació de video basats en regions. En concret, es mostra com es pot utilitzar un metodologia de 'rate-distortion' en aquest tipus de problemes. El segon problema que es tracta és com introduir funcionalitats basades en el contingut en un d'aquests sistemes de codificació de video.

La teoria de 'rate-distortion' defineix l'optimalitat en la codificació com la representació d'un senyal que, per una taxa de bits donada, resulta en una distorsió mínima al reconstruir el senyal. En el cas de sistemes de codificació basats en regions, això implica obtenir una partició òptima i al mateix temps, un repartiment òptim dels bits entre les diferents regions d'aquesta partició. Aquest problema es formalitza per sistemes de codificació no escalables i es proposa un algorisme per solucionar-lo. Aquest algorisme s'aplica a un sistema de codificació concret anomenat SESAME. En el SESAME, cada quadre de la seqüència de video es segmenta en un conjunt de regions que es codifiquen de forma independent. La segmentació es fa seguint criteris d'homogeneitat espaial i temporal. Per eliminar la redundància temporal, s'utilitza un sistema predictiu basat en la informació de moviment tant per la partició com per la textura. El sistema permet seguir l'evolució temporal de cada regió per tota la seqüència. Els resultats de la codificació són òptims (o quasi-òptims) pel marc donat en un sentit de 'rate-distortion'. El procés de codificació inclou trobar una partició òptima i també trobar la tècnica de codificació i nivell de qualitat més adient per cada regió.

Més endevant s'investiga el problema de codificació de video en sistemes amb escalabilitat i que suporten funcionalitats basades en el contingut. El problema es generalitza incloent en l'esquema de codificació les dependències espaials i temporals entre els diferents quadres o entre les diferents capes d'escalabilitat. En aquest cas, la solució requereix trobar la partició òptima i les tècniques de codificació de textura òptimes tant per la capa base com per la capa de millora. A causa de les dependències que hi ha entre aquestes capes, la partició i el conjunt de tècniques de codificació per la capa de millora dependran de les decisions preses en la capa base. Donat que aquest tipus de solucions generalment són molt costoses computacionalment, també es proposa una solució que no té en compte aquestes dependències.

Els algorismes obtinguts s'apliquen per extendre SESAME. El sistema de codificació extès, anomenat XSESAME suporta diferents tipus d'escalabilitat (PSNR, espaial i temporal) així com funcionalitats basades en el contingut i la possibilitat de seguiment d'objectes a través de la seqüència de video. El sistema de codificació permet utilitzar dos modes diferents pel que fa a la selecció de les regions de la partició de la capa de millora: El primer mode (supervisat) està pensat per utilitzar funcionalitats basades en el contingut. El segon mode (no supervisat) no suporta funcionalitats basades en el contingut i el seu objectiu és simplement obtenir una codificació òptima a la capa de millora.

Un altre tema que s'ha investigat és la integració d'un mètode de seguiment d'objectes en el sistema de codificació. En el cas general, el seguiment d'objectes en seqüències de video és un problema molt complex. Si a més aquest seguiment es vol integrar en un sistema de codificació apareixen problemes addicionals degut a que els requisits necessaris per obtenir eficiència en la codificació poden entrar en conflicte amb els requisits per una bona precisió en el seguiment d'objectes. Aquesta aparent incompatibilitat es soluciona utilitzant un enfoc basat en una doble partició de cada quadre de la seqüència. La partició que s'utilitza per la codificació es resegmenta utilitzant criteris purament espaials. Al projectar aquesta segona partició permet una millor adaptació dels contorns de l'objecte a seguir. L'excés de regions que implicaria aquesta re-segmentació s'elimina amb una etapa de fusió de regions realitzada *a posteriori*.

Resumen

En este trabajo se estudia el problema de la compresión de vídeo utilizando funcionalidades basadas en el contenido en el marco teórico de los sistemas de codificación de secuencias de vídeo basados en regiones. Se tratan básicamente dos problemas: El primero está relacionado con la obtención de una codificación óptima en sistemas de codificación de vídeo basados en regiones. En concreto, se muestra como se puede utilitzar un metodología de 'rate-distortion' para este tipo de problemas. El segundo problema tratado es como introducir funcionalidades basadas en el contenido en uno de estos sistemas de codificación de vídeo.

La teoría de 'rate-distortion' define la optimalidad en la codificación como la representación de una seal que, para un tasa de bits dada, resulta en una distorsión mínima al reconstruir la seal. En el caso de sistemas de codificación basados en regiones, esto implica obtener una partición óptima y al mismo tiempo, un reparto óptimo de los bits entre las diferentes regiones de esta partición. Este problema se formaliza para sistemas de codificación no escalables y se propone un algoritmo para solucionar este problema. Este algoritmo se aplica a un sistema de codificación concreto llamado SESAME. En el SESAME, cada cuadro de la secuencia de vídeo se segmenta en un conjunto de regiones que se codifican de forma independiente. La segmentación se hace siguiendo criterios de homogeneidad espacial y temporal. Para eliminar la redundancia temporal, se utiliza un sistema predictivo basado en la información de movimiento tanto para la partición como para la textura. El sistema permite seguir la evolución temporal de cada región a lo largo de la secuencia. Los resultados de la codificación son óptimos (o casi-óptimos) para el marco dado en un sentido de 'rate-distortion'. El proceso de codificación incluye encontrar una partición óptima y también encontrar la técnica de codificación y nivel de calidad más adecuados para cada región.

Más adelante se investiga el problema de la codificación de vídeo en sistemas con escalabilidad y que suporten funcionalidades basadas en el contenido. El problema se generaliza incluyendo en el esquema de codificación las dependencias espaciales y temporales entre los diferentes cuadros o entre las diferentes capas de escalabilidad. En este caso, la solución requiere encontrar la partición óptima y las técnicas de codificación de textura óptimes tanto para la capa base como para la capa de mejora. A causa de les dependencias que hay entre estas capas, la partición y el conjunto de técnicas de codificación para la capa de mejora dependerán de las decisiones tomadas en la capa base. Dado que este tipo de soluciones

generalmente són muy costosas computacionalmente, también se propone una solución que no tiene en cuenta estas dependencias.

Los algoritmos obtenido se usan en la extensión de SESAME. El sistema de codificación extendido, llamado XSESAME soporta diferentes tipos de escalabilidad (PSNR, espacial y temporal) asi como funcionalidades basadas en el contenido y la posibilidad de seguimiento de objectos a través de la secuencia de vídeo. El sistema de codificación permite utilizar dos modos diferentes por lo que hace referencia a la selección de les regiones de la partición de la capa de mejora: El primer modo (supervisado) está pensado para utilitzar funcionalidades basadas en el contenido. El segundo modo (no supervisado) no soporta funcionalidadess basadas en el contenido y su objectivo es simplemente obtener una codificación óptima en la capa de mejora.

Otro tema investigado es la integración de un método de seguimiento de objetos en el sistema de codificación. En el caso general, el seguimiento de objetos en secuenciass de vídeo es un problema muy complejo. Si este seguimiento se quiere integrar en un sistema de codificación aparecen problemas adicionales debido a que los requisitos necesarios para obtener eficiencia en la codificación pueden entrar en conflicto con los requisitos para obtener una bona precisión en el seguimiento de objetos. Esta aparente incompatibilidad se soluciona usando un enfoque basado en una doble partición de cada cuadro de la secuencia. La partición que se usa para codificar se resegmenta usando criterios puramente espaciales. Al proyectar

Proyectando esta segunda partición se obtiene una mejor adaptación de los contornos al objeto a seguir. El exceso de regiones que implicaría esta resegmentación se elimina con una etapa de fusión de regiones realizada *a posteriori*.

Abstract

This work addresses the problem of video compression with content-based functionalities in the framework of segmentation-based video coding systems.Two major problems are considered. The first one is related with coding optimality in segmentation-based coding systems. Regarding this subject, the feasibility of a rate-distortion approach for a complete region-based coding system is shown. The second one is how to address content-based functionalities in the coding system proposed as a solution of the first problem.

Optimality, as defined in the framework of rate-distortion theory, deals with obtaining a representation of the video sequence that leads to a minimum distortion of the coded signal for a given bit budget. In the case of segmentation-based coding systems this means to obtain an 'optimal' partition together with the best coding technique for each region of this partition so that the result is optimal in an operational rate-distortion sense. The problem is formalized for independent, non-scalable coding. An algorithm to solve this problem is provided as well.

Ths algorithms is applied to a specific segmentation-based coding system, the so called SESAME. In SESAME, each frame is segmented into a set of regions, that are coded independently. Segmentation involves both spatial and motion homogeneity criteria. To exploit temporal redundancy, a prediction for both the partition and the texture of the current frame is created by using motion information. The time evolution of each region is defined along the sequence (time tracking). The results are optimal (or near-optimal) for the given framework in a rate-distortion sense. The definition of the coding strategy involves a global optimization of the partition as well as of the coding technique/quality level for each region.

Later, the investigation is also extended to the problem of video coding optimization in the framework of a scalable video coding system that can address content-based functionalities. The focus is set in the various types of content-based scalability and object tracking. The generality of the problem has also been extended by including the spatial and temporal dependencies between frames and scalability layers into the optimization schema. In this case the solution implies finding the optimal partition and set of quantizers for both the base and the enhancement layers. Due to the coding dependencies of the enhancement layer with respect to the base layer, the partition and the set of quantizers of the enhancement layer depend on the decisions made on the base layer. Also, a solution for the independent opti-

mization problem (i.e. without tacking into account dependencies between different frames of scalability layers) has been proposed to reduce the computational complexity.

These solutions are used to extend the SESAME coding system. The extended coding system, named XSESAME, supports different types of scalability (PSNR, Spatial and temporal) as well as content-based functionalities, such as content-based scalability and object tracking. Two different operating modes for region selection in the enhancement layer have been presented: One (supervised) aimed at providing content-based functionalities at the enhancement layer and the other (unsupervised) aimed at coding efficiency, without content-based functionalities.

Integration of object tracking into the segmentation-based coding system is also investigated. In the general case, tracking is a very complex problem. If this capability has to be integrated into a coding system, additional problems arise due to conflicting requirements between coding efficiency and tracking accuracy. This is solved by using a double partition approach, where pure spatial criteria are used to re-segment the partition used for coding. The projection of the re-segmented partition results in more precise adaption to object contours. A merging step is performed *a posteriori* to eliminate the excess of regions originated by the re-segmentation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, the importance of visual communications is increasing steadily. With multitude of applications and services, the high bandwidth consumption of this kind of communication and the limited bandwidth resources available make necessary the use of efficient coding methods. In addition, many of these new services require more sophisticated representations of data in order to allow end-user interactivity and content-based functionalities.

In this context, the problem of how to represent images in a suitable way for storage or transmission has been widely addressed in recent years. It is well known that the representation of the canonical form of a video sequence requires a very large number of bits making impractical its use in several applications. Moreover, the canonical representation is not well suited to deal with these new interactive and content-based services.

Thus, a wide variety of video coding techniques have been developed. At first, they aimed only at a reduction of the amount of bits necessary for the representation of the video sequence. In a second step, new coding techniques are being developed taking into account the semantic meaning of the objects represented in the video sequence.

In this case the goal is to drive the coding process taking into account the fact that the video sequence can be viewed as a representation of a set of meaningful objects, and to code separately each object along the video sequence, so that it is possible to manipulate these objects at the decoder side.

Most of the existent coding methods deal with compression efficiency by exploiting spatial and temporal statistical redundancies between the pixels of the sequence. Lossy coding schemes are used in most applications because perfect reconstruction techniques do not give high enough compression gains. The most successful methods use a hybrid approach where some kind of waveform coding is used to reduce spatial redundancy, while predictive coding followed by lossy coding of the prediction error attempts a similar reduction in the tempo-

ral domain. The efforts have resulted in many different techniques and some video coding standards.

Nowadays, all these video coding techniques can be inscribed into two main categories, according to its envisioned application. The first category basically aims at coding efficiency while the second one at providing interactivity and content-based functionalities.

The methods that follow the first approach use a structure for coding that is imposed by the canonical representation of the video sequence. Thus, the entities to be coded are the pixels or fixed-shape blocks of pixels. Such a regular and prefixed structure allows a simple but powerful modeling of the data. Many coding techniques and standards have been developed according to this paradigm. Among them, it can be noticed JPEG for still images and H261, H263, MPEG-1, MPEG-2 for video sequence coding. For a complete review of these standards see [43, 44, 17, 81, 126]

However, the simple representation model used by these methods presents many problems. In most cases image data sources are neither stationary nor ergodic, so the pixel based or block based representation fails at effectively removing the redundancy present in the video sequences. To overcome these limitations, a set of techniques have been proposed [55] that use representation models that adaptively match the local statistics of the image, thus allowing the coding system to follow more closely the variations of the video source. As a result, the basic units subject to the coding process are not pixels or fixed-shape blocks of pixels. Instead, higher level entities are defined by an analysis and/or segmentation process.

The second approach is intended to provide content-based functionalities. It has emerged in the last years because the methods designed only for coding efficiency could not fulfill the requirements of new multimedia applications, where a higher level representation of the visual data is required to allow the access and manipulation of video objects. Currently, MPEG-4 (and JPEG2000 [132] for still image coding) is the only established standard that follows this approach.

In this context, while coding efficiency is still a requirement, it is not the most important. For this reason, it has been seen that the more flexible segmentation-based representation is better suited to represent objects with some semantic meaning. It this sense, these techniques can be appropriate for the new coding standards, such as MPEG-4, where content-based functionalities are essential.

As the regions that result after a segmentation process in a segmentation based coding system (SBCS) can have very different source statistics, there is no single coding technique that can perform equally well on all of these regions. In this case, the coding efficiency can be improved by coding each region with its optimal coding technique. Because both the coding cost and the visual quality associated with each coding technique will be different, the decision on which technique to use for each region is in fact, an optimization problem.

Optimality can be defined in the framework of rate-distortion theory [7, 37], where the

problem of bit allocation can be stated as obtaining a representation of the sequence that leads to a minimum distortion of the coded signal for a given bit budget. For block-based systems, this can be accomplished by choosing the appropriate quantizer for each frame or for each block in the frame. In the case of segmentation-based coding systems, the problem is far more difficult because the goal is to obtain an 'optimal' partition together with the best coding technique for each region of this partition so that the result is optimal in a rate-distortion sense. That is, an optimal distribution of the given bit budget among a set of regions.

In practice, this problem is tackled in an operational framework [91]; in this case the search for optimality is restricted by fixing *a priori* the set of coding techniques and the set of partitions. The result is the best achievable solution for this given framework.

To complete this picture of the video coding process, it is to be noted that coding efficiency or content-based functionalities are not the only point to be addressed. To face today's broad range of applications such as TV and HDTV broadcasting, Internet, Video on demand, network interoperability, etc. other services are necessary. In many video coding applications, it is required that the encoding process is performed in such a way that the resulting bitstream can be decoded by receivers with different display capabilities. To avoid the overhead of different bitstreams for each different kind of end-user request, video coding scalability aims at generating two or more video layers from a single video source in an incremental way. This is, the so called *base layer*, is encoded by itself to provide a basic representation of the image, while the other layers (*enhancement layers*), when added progressively to the base layer, produce increasing quality of the reconstructed signal. This functionality is useful for applications where the receiver display or a complete network system are either not capable or not willing to display or transmit the full resolution supported by all the layers.

1.2 Thesis organization

This thesis addresses the problem of optimal video scalability in the framework of a segmentation-based coding system with support for content-based functionalities. This problem implies:

- Obtaining the optimal partition for each scalability layer.
- Finding the best bit allocation for the resulting regions of the different layers.
- Ensuring that the whole process is done in such a way that the resulting system can provide content-based functionalities.

A solution is proposed for this problem, which is applied to improving a coding system capable to describe the scene in terms of regions that can be coded independently.

It is to be noted that, even if this solution provides results that are optimal in a rate-distortion sense, this optimality is only ensured for the framework being chosen, in this case, a

segmentation-based coding system. While other frameworks may result in better compression efficiency, this particular one has been chosen because it provides a very versatile representation model that is very well suited to the implementation of content-based functionalities.

The proposed algorithms can construct scaled layers from a video sequence, each one with either fixed bit-rate or fixed quality. These layers are composed of regions at different resolution levels. The base layer consists on a basic representation of the whole image, while the enhancement layers will code selected regions or objects as separate entities, improving its coded quality, and also allowing content based manipulation. The selection of the set of regions that will be coded at a given layer can be done in a *supervised* or *unsupervised* mode.

Note that, in this context, regions and objects are not equivalent concepts. Regions are the basic high-level entities subject to coding in a segmentation based coding system when coding efficiency is the goal of the process. These regions are defined by a segmentation step according to a homogeneity criterion and they may not have any semantic meaning. Objects are entities with a semantic meaning. They are defined by an analysis process. In this work, an object can have an internal structure and be composed of several regions. This internal structure results from the segmentation-based nature of the coding algorithm. This way, objects can be treated as regions or groups of regions. Therefore, content-based functionalities can be addressed in a natural way.

This thesis is organized in three parts:

Part I presents the fundamental concepts as well as the state of the art on video coding techniques focusing on established coding standards. Chapter 2 is devoted to introduce concepts that will be used along all the work. Section 2.2 introduces the concept of scalability and presents an overview of the different scalability types, analyzing them from the segmentation-based point of view. In Section 2.3 the concept of optimal coding is discussed using a rate-distortion approach; a brief review is given on how this optimization approach has been previously addressed for video coding. Chapter 3 presents the state of the art on video coding techniques aimed at coding efficiency, focusing on established coding standards. To help understanding the current video coding scenario, Section 3.1 is a brief presentation of block-based techniques and established standards. Section 3.2 is devoted to show the general concepts of segmentation based coding techniques in its original sense; this is, having coding efficiency as the main goal. Chapter 4 presents an overview of video coding systems that can address content-based functionalities.

Part II is devoted to the study of optimal strategies for coding efficiency in a segmentation-based coding system. In Chapter 5 the first original contributions of this thesis are presented. First, Section 5.2 is devoted to properly formulate the different problems to be solved and the proposed solutions. In Chapter 6, a description of the coding system (SESAME) in its basic form is given. This coding system was presented as a proposal for the MPEG-4 Verification Model. While the author of this thesis participated actively in its development

[16, 82, 97, 73, 117], the whole system is the result of the work of many people at different laboratories (UPC, LEP and CMM). In its basic state it was aimed at coding efficiency, but its ability to address content-based functionalities was early noticed [16].

Part III is devoted to the study of optimal strategies for content-based functionalities in a segmentation-based coding system. This part consists almost entirely on original contributions. In Chapter 7 the problem of video coding optimization is tackled in the framework of a system that can address content-based functionalities. Chapter 8 presents the modifications introduced in the basic coding system in order to support the different types of scalability and to properly address content-based functionalities. Section 8.11 shows an example of content-based functionalities (object tracking). Experimental results are provided in Chapter 9. In Section 9.4 there is a comparison between dependent and independent optimization.

Finally, Chapter 10 provides some conclusions and proposes future extensions to this work.

Part I

Background and state of the art

Chapter 2

Video coding

2.1 Introduction

In this chapter, a brief overview of some of the most important video coding techniques and established standards will be given, with special emphasis on segmentation-based coding techniques. The purpose is not to give a detailed view of video coding, but to establish the context for this thesis by stating the main points that are to be addressed and their existing approaches. The reader is referred to [134, 24, 126] in order to have a deeper view on the general topic.

Temporal prediction versus 3D extensions

Multitude of solutions have been proposed for the problem of video coding. The common point in almost all of them is that they aim at removing redundancy in both the temporal as well as spatial dimensions. This can be accomplished in a variety of manners, but the most popular and successful one is to use some form of predictive coding in order to remove the temporal redundancy. This prediction can be done in different ways (i.e in the sub-band or transform domain or directly in the signal domain). Other approaches, such as extending the 2-D techniques available for still images to a 3-D space have been investigated in the literature [88, 109, 49, 59], but in general they introduce time artifacts and high delay. Moreover, they are less efficient due to the little correlation that exists in the presence of motion between samples occupying the same position in consecutive images [31]. We will work with temporal prediction.

Intra-frame and Inter-frame modes

In almost any coding scheme, two modes of transmission can be distinguished: intra-frame and inter-frame [116]. The intra-frame mode consists on sending the information of each frame independently. This is useful for still images, for the first frame of the sequence

or to refresh periodically the information on the receiver. By contrast, the inter-frame mode relies on the characterization and coding of the time evolution of the partition from one frame to the next one.

Multitude of methods can be used to remove spatial redundancy. They can be applied to the original image (intra-mode coding) or to the difference between the predicted and the original image if motion compensation is used (inter-mode coding). Usually these methods try to compact most of the energy of the signal into a few decorrelated frequency coefficients. Then, these coefficients are quantized and some kind of entropy coding is applied to the result, usually Huffman or arithmetic coding.

Temporal prediction in inter mode

All the here-reviewed techniques use motion compensation (MC) as the way to reduce the temporal correlation between consecutive frames of a video signal. Motion compensation makes use of the motion field of a video signal at a given instance to predict following instances. This process is known as Motion Estimation (ME). An in-depth study of motion estimation can be found in [129] and [141]. By using past frames (predictive) or past and future frames (bidirectional), motion compensation techniques construct a prediction of each frame of the sequence. Then, a lossy coding of the resulting error images is performed.

Different approaches exist to compute the motion information:

- Block motion compensation [129, 87]. These techniques partition the image into small blocks and assign a translational displacement to each of these blocks by minimizing a disparity measure. They require low overhead information and are easily implementable in hardware. Many systems use these techniques, for example MPEG1, MPEG2 and H263.

- Region motion compensation [21]. In segmentation-based coding systems, the motion information can be estimated independently for each region. Usually, each region is characterized by a coherent motion and more complex motion models can be used. If the segmentation criterion is motion homogeneity, the accuracy of the estimation can be highly increased with respect to the block-based case. Usually the complexity of this approach is higher than the block-based counterpart.

- Deformable mesh [40, 19, 136]. In this case, the image is modeled as a 2D-mesh structure. The image is partitioned without any a priory knowledge like in block based systems but here the nodes of the mesh rather than the blocks are used to estimate the vectors. Triangular or rectangular mesh structures are used. The mesh nodes are tracked along the sequence, and pixel motion in the mesh patches is interpolated using affine warping according to the motion of surrounding nodes.

Coding versus functionalities

Regardless of the previous choices (although sometimes related), two main categories of coding approaches will be considered: the first one aims at coding efficiency. That is, at providing the best quality for the reconstructed signal with the minimum quantity of bits to be transmitted/stored. The methods falling into this category are usually block-based, even though some region-based methods can also be included in this category.

The second category is that of content-based methods. These methods use objects with a semantic meaning as the primitives of the coding (instead of blocks of pixels or regions of the image) in order to handle content-based functionalities.

Content-based coding systems can be constructed by using a segmentation-based paradigm (i.e. SESAME) or by adapting block-based techniques as in MPEG-4.

Scalability

No matter which category is considered, many applications require that the same bit-stream is to be received by both high quality and low cost decoders (for example HDTV and SDTV). In other cases, independently of the decoder performance, it is desirable to have the possibility to quickly browse and preview a coarse version of the video without the overhead associated to a complete decoding of the bitstream.

Optimization

Typical video encoders encompass a mechanism that controls the trade-off between the picture quality and the bit-rate. In many cases, video encoders are required to provide streams with fixed bit-rate (or fixed quality), leading to a bit-allocation problem. Such restrictions are specially important in scalable coders. Besides, these systems pose the problem of how to construct the scalable layers in such a way that the result is globally optimal (this is, given a total bit budget for all the layers, how to obtain the maximum global quality).

Multiple solutions have been developed to provide methods of bit-allocation. As all the considered coding systems are lossy in nature, bit-allocation methods work by adapting *locally* (i.e at the block or macro block level in block-based coding systems or at the region level in segmentation-based coding systems) the encoding parameters to the statistics of the source, so that R-D optimality (the best quality for a given rate or the lowest rate for a given quality) is ensured. This challenge is compounded by the fact that video sequences contain widely varying content and motion. The Rate-Distortion theory [123, 7] provides the mathematical basis for the bit allocation methods in the data compression field.

The work reported in this thesis is mainly about scalability in content-based coding systems. Thus, in this chapter the different types of scalability will be presented, as well as an analysis of its role in object-based coding systems. In this work, rate-distortion techniques

are applied to construct the scaled layers in an optimal way. This chapter also provides the foundations of the Rate-Distortion techniques that are to be used in the following chapters.

2.2 Scalability

According to [126], the intention of scalable coding is to provide interoperability between different services and to flexibly support receivers with different display capabilities. Receivers either not capable or not willing to reconstruct the full resolution video can decode subsets of the layered bit stream to display video at lower spatial or temporal resolution or with lower quality.

A multi-scale representation can be achieved by downscaling the input video signal into a lower resolution video (down-sampling in the spatial or temporal domains). The down-scaled version is encoded into a base-layer bitstream with reduced bit-rate. The up-scaled reconstructed base-layer video (up-sampled spatially or temporally) is used as a prediction for the coding of the original input video signal. The prediction error is encoded into an enhancement-layer bitstream. If the receiver is either not capable or not willing to display the full quality video, a downscaled video signal can be reconstructed by only decoding the base-layer bitstream. It is important to notice, however, that the display of the video at the highest resolution with reduced quality is also possible by only decoding the lower bit-rate base layer.

Currently, many video coding standards support scalability. For example, in MPEG-2 there are three scalable modes, based on coding each layer with different spatial, frequential or temporal resolutions [126]. All these techniques have in common the fact of dealing with the image at the block level. In addition, they aim at increasing the quality of the full frame in the enhancement layer (*Frame scalability*).

New standards allowing content-based functionalities, as MPEG-4 [125], put other requirements on scalability. In these coding schemes, the partition of the sequence into semantic-based objects, also named Video Objects, require new scalability techniques that can operate with individual objects of arbitrary shape in a natural way (*Object scalability*), maintaining as much as possible the coding efficiency. These techniques must be able to structure scalability layers on an object selection basis, rather that on pixel or block basis. In this work, *Object scalability* refers to a mode where in the enhancement layers all the available bit budget is spent on the coding of a selected object or objects, improving its coded quality, and allowing content based functionalities such as independent retrieval, tracking and manipulation of the objects. In this case, the subjects of the coding are not full frames but Video Objects. On the other side, *Frame scalability* refers to a mode where the bit budget on the enhancement layer is used to improve the coding quality of the full frame. This mode can be considered a particular case of *Object scalability* where the selected object is the full frame.

In segmentation-based video coding systems, the fact that the scene is described in terms of regions, makes possible defining a hierarchical set of partitions of the image at different levels of detail [97]. This is, a set of partitions that range from a coarse representation in

the first level to a fine representation in the last level, in various steps. Each level can be constructed by splitting regions present in the previous level, so that the contours of the regions in previous levels are preserved.

Using this hierarchical partition, the scalability layers can be constructed in such a way that different regions or objects are coded in different scalability layers. While the base layer provides a representation of all the image, an appropriate method allows the selection (in a supervised or unsupervised way) of the regions or objects that are coded in the enhancement layers.

For simplicity, we will consider only the case of two layers: a base and an enhancement layer. The extension to n-layers scalability is straightforward.

2.2.1 Temporal scalability

When coding video sequences, specially at low bit-rates, a common approach is to decrease the coding frame rate. At the decoder, the frames that have not been encoded can be created by repeating the last encoded frame. This procedure obviously has the effect of decreasing the temporal resolution of the decoded sequence.

In temporal scalability, the base layer is coded at a low temporal resolution to provide a low bit-rate representation of the video sequence. In the enhancement layers, temporal resolution is improved by coding frames that were skipped in the base layer (see Figure 2.1).

Figure 2.1: Temporal Scalability: Middle row shows the frames encoded at the base layer. Top row shows the frames encoded at the enhancement layer. Bottom row shows the reconstructed sequence using base plus enhancement layers

In some systems (i.e. MPEG-2 [126]) stereoscopic video can be supported through temporal scalability. In this case, layering is achieved by providing a prediction of one of the images of the stereoscopic video (i.e. left view) in the enhancement layer based on coded images from the opposite view transmitted in the base layer.

2.2.2 PSNR scalability

In PSNR scalability (see Figure 2.2) the enhancement layer improves the coding of the current frame by refining the texture of the base layer. This feature provides graceful degradation of the video quality in prioritized transmission media. It is also useful for browsing purposes. If only the base layer is decoded, a low bit-rate, low quality version of the signal is decoded.

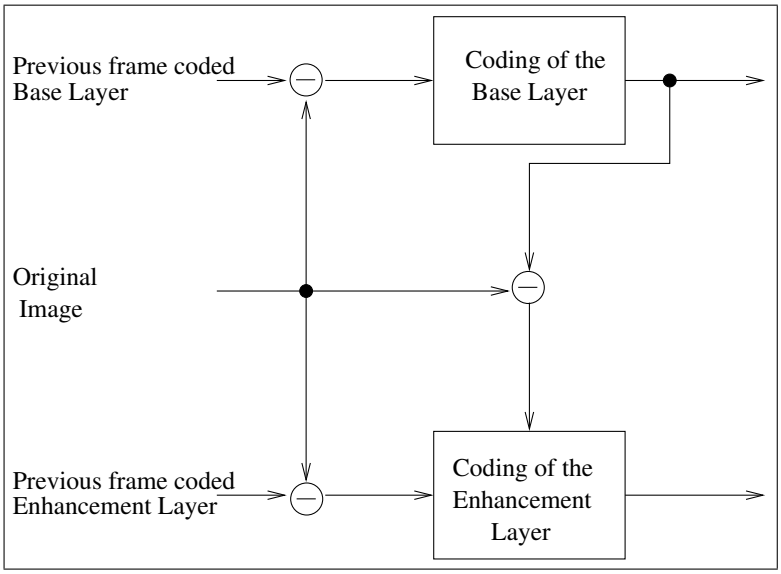

Figure 2.2: PSNR scalability

2.2.3 Fine-grained scalability (FGS)

FGS [100] allows small quality steps by adding or deleting layers of extra information with fine granularity (quality scalability).

It is built to allow a separation between the encoding and the distribution process: The encoder produces only one (rather large) enhancement layer bitstream, while the FGS server application truncates this bitstream according to the characteristics of the transmission channel (e.g. bit rate) or the decoder (e.g. computational power).

While in the SNR scalability technique described in Section 2.2.2 the reconstruction error is requantized and coded using the same mechanisms as the base layer, in FGS the reconstructed

error of the base layer is encoded in the enhancement layer using a *bit plane* representation of the texture coefficients. The coefficient information is grouped bit plane by bit plane, starting from the most significant bit plane (MSB), until the least significant bit plane (LSB). Then, the MSB of the enhancement layer signal are encoded into the bitstream, followed by the second most significant bitplanes, nd so on. Thus, it is possible to stop the transmission of the enhancement layer data at any point, while being able to make use of all the transmitted data up to that point.

FGS is useful in a number of environments, notably for streaming purposes, but also for dynamic (statistical) multiplexing of preencoded content in broadcast environments.

2.2.4 Spatial scalability

Spatial scalability (see Figure 2.3) is very similar to PSNR scalability. The only difference is that the scalability layers are coded at different resolution levels. It has been developed to support displays with different spatial resolutions at the receiver. This is the case, for example of HDTV/TV systems.

It is usually based on pyramidal coding [11]. In the base layer, a sub-sampled version of the original frame is coded. The coding residue is obtained from the original image at full resolution and an up-sampled version of the reconstructed base layer.

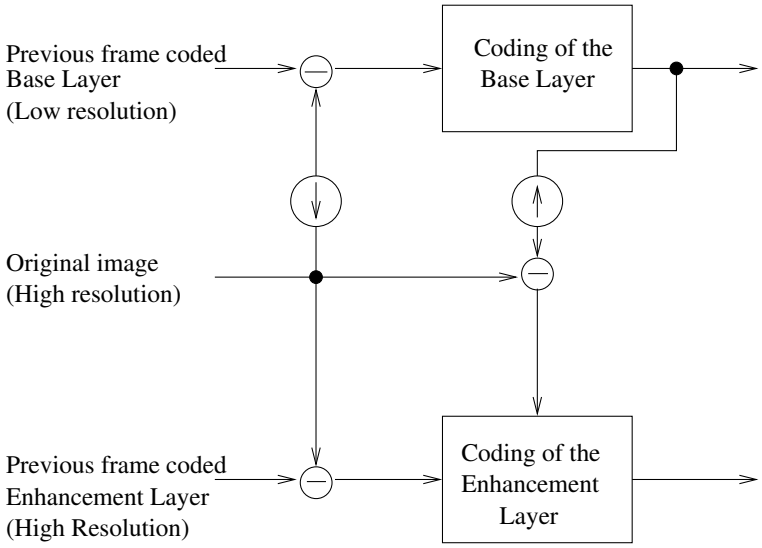

Figure 2.3: Spatial scalability

2.2.5 Object scalability

As mentioned previously, *Object scalability* refers to a mode where in the enhancement layers all the available bit budget is spent on the coding of a selected object or objects, improving its coded quality. Therefore, the previous methods can be adapted to be object-based. Figure 2.4 shows an example of Spatial object scalability in a segmentation-based coding system. In the base layer, the scene is partitioned into a set of regions that are coded independently. In the enhancement layer, the coding of each frame is improved by re-segmenting the regions and by refining the texture of these regions but only inside the selected object, in this case, the two anchor people (Figure 2.4 shows only base and enhancement layer partitions, not the coded images).

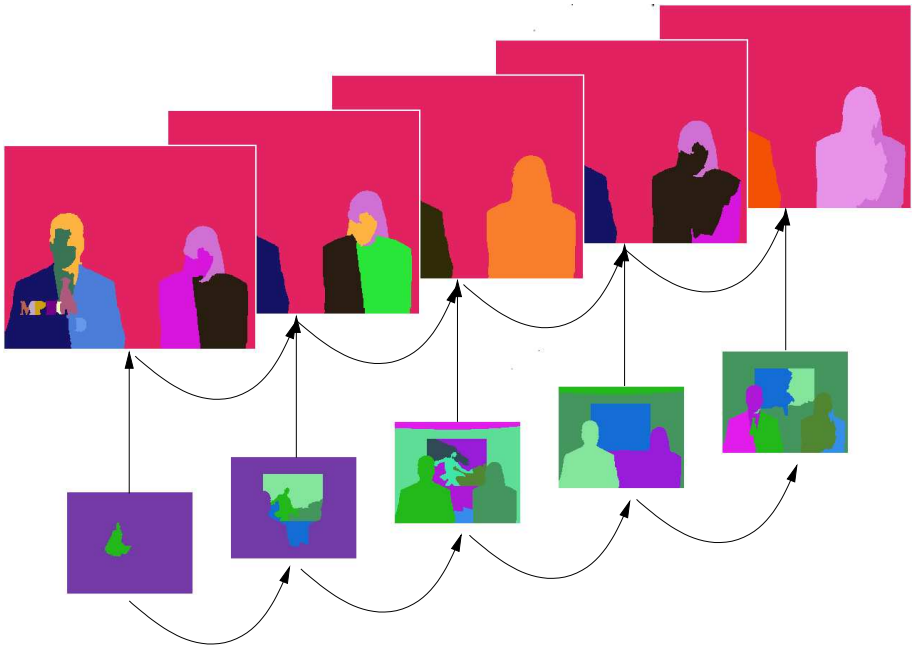

Figure 2.4: Spatial object scalability

In temporal object scalability, only the selected object is to be coded in the enhancement layer. The rest of the image (background) is copied at the decoder from a background memory of the previous frame base layer. In the case of moving or shape varying objects, temporal object scalability has to deal with the possible apparition of uncovered background zones (see Figure 2.5). This occurs when the shape or the size of the object coded in the enhancement layer of frame #(n+k) is different from its reference in the base layer of frame #(n). In this case, the coded object will not fit exactly in the hole left in the base layer.

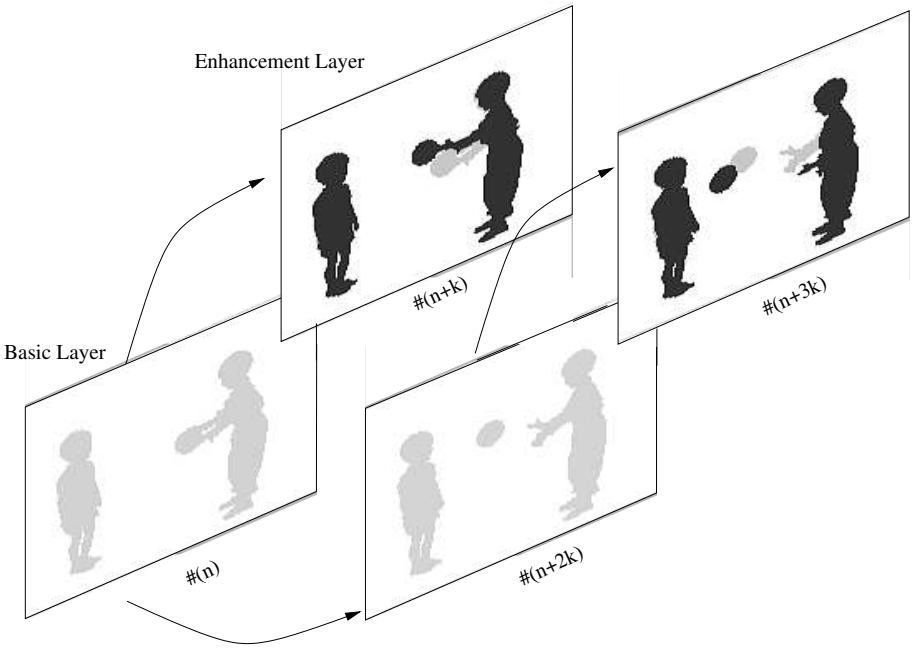

Figure 2.5: Temporal object scalability

2.2.6 Final comment

This work addresses the problem of optimal video scalability in the framework of a segmentation-based coding system with support for content-based functionalities. In Chapter 8 it will be shown how the different types of scalability (temporal, PSNR and spatial scalability, in both the full-frame and object-based approaches) can be implemented on a SBCS.

2.3 Bit Allocation and Rate-Distortion theory

Today's coding systems decompose the input information into signal units (i.e. frames, blocks, regions, ...) that are coded separately. This decomposition can be done spatially (the different regions or rectangular blocks in a frame) and/or temporally (the different frames in a sequence). Most video coding systems use "lossy" compression approaches because much higher compression ratios are possible. In these systems, the encoder can normally choose between a set of parameters in order to adjust the desired quality or bit-rate of the encoded signal. This results in a trade-off between cost and quality, also referred as rate-distortion trade-off. For instance, in transform coding, the quantization degree of the transform coefficients is an example of an encoder parameter.

As the signal is decomposed into signal units, it may happen that different units have different statistical properties. In this case, it will be beneficial to be able to adjust the rate-distortion trade-off separately for each signal unit so that, at the end, the result is globally optimal.

The selection of parameters can be done at different levels, depending on the actual coding system. For instance, some encoders can select parameters at the block or region level, while others can only select the parameters at the frame level. The ability of selecting parameters at a "lower" level adds complexity to the encoder but greatly improves its ability to adapt from deviations of the underlying statistical model.

The problem of finding the "best" parameters is referred usually as bit-allocation, because it is stated usually as finding the best distribution of a given bit-budget among all the signal units so that the final quality of the decoded signal is the best possible (this is, distortion is minimized).

Another problem related to the previous one arises from the fact that usually the encoder should balance the quality of the decoded images with the channel capacity. Rate Control (RC) is a decision making process where the desired encoding rate for a source video can be met accurately by properly setting a set of encoder parameters. Most rate control methods usually include the bit-allocation step so that the bit-rate of the encoded signal is adapted to the channel capacity and simultaneously an optimal distribution of this bit-rate among the signal units is performed.

Optimality can be studied in the framework of Shannon's rate-distortion theory [37] from an analytical point of view that seeks to provide performance bounds for the cost-quality trade-off. The problem of this approach is that it is very difficult to find closed solutions except for a few particular cases because it requires an accurate source characterization. Moreover, the results it provides are non-constructive.

A more practical method is to use a less general approach and instead of finding the "best" system, restrict the problem to achieving optimality for a given system. This is, the

rate-distortion approach is not used to model the framework for a coding scheme, but to find the best operating points for that specific scheme.

The operational approach to the problem of minimizing the distortion D of independent signal blocks subject to a restriction on the coding cost R has been widely addressed in source coding literature [124, 105, 107].

An optimal solution to this problem can be obtained via dynamic programming [6], but this approach leads to a very high computational complexity, so it is not appropriate for video sequence coding. Shoham and Gersho [124] solved the problem of optimal bit allocation for independent signal blocks and an arbitrary set of quantizers, using a fast algorithm based on the Lagrange-multiplier optimization method.

The constrained problem can be formulated as: Given an image to code $\mathcal{I}$, a coding bit budget R_{budget}, a partition $\mathcal{P}$ of $\mathcal{I}$, and an available set of quantizers $\mathcal{Q}$, $\forall$ region $P_i \in \mathcal{P}$ find the mapping (P_i, Q_j) such that:

$$\min_{Q_j \in \mathcal{Q}} \sum_i D_i(Q_j) \quad \text{subject to} \quad \sum_i R_i(Q_j) \leq R_{budget} \tag{2.1}$$

where $R_i(Q_j)$ and $D_i(Q_j)$ are, respectively, the rate and distortion associated to the region i when using quantizer j.

This constrained problem can be converted into an unconstrained problem combining the distortion and the rate by means of a Lagrange multiplier λ, and then, minimizing the Lagrangian cost function $J(\lambda) = D + \lambda R$. This unconstrained problem is much easier to solve, and fast algorithms can be used. The unconstrained problem can be formulated as:

$$\min_{Q_j \in \mathcal{Q}} \sum_i (D_i(Q_j) + \lambda R_i(Q_j)) \qquad \lambda \in \Re, \quad \lambda \geq 0 \tag{2.2}$$

At optimality, λ is the same for all the signal blocks. As the signal blocks are independent, the problem can be solved separately for each block. Thus, equation 2.2 can be expressed as:

$$\sum_i \min_{Q_j \in \mathcal{Q}} (D_i(Q_j) + \lambda R_i(Q_j)) \tag{2.3}$$

If a $\lambda^* = \lambda^*(Q_1, \ldots Q_N)$ can be found such that $R_T(\lambda^*) = \sum R_i = R_{budget}$ the constrained and unconstrained problems are equivalent and a solution has been found. We have to note that this formulation does not guarantee to find always the solution to the constrained problem. Some times there is no λ^* such that $\sum_i R_i(\lambda^*) = R_{budget}$. In these cases, a near-to-optimal solution is found. It is possible to demonstrate that R(λ) is monotonically non-increasing with λ; that is, if $\lambda_2 \geq \lambda_1$, then $R(\lambda_2) \leq R(\lambda_1)$. This allows to determine the value of λ^* using an iterative fast convex search algorithm (bisection or Newton's method). This method traces out the points that are on the convex hull of all possible rate-distortion

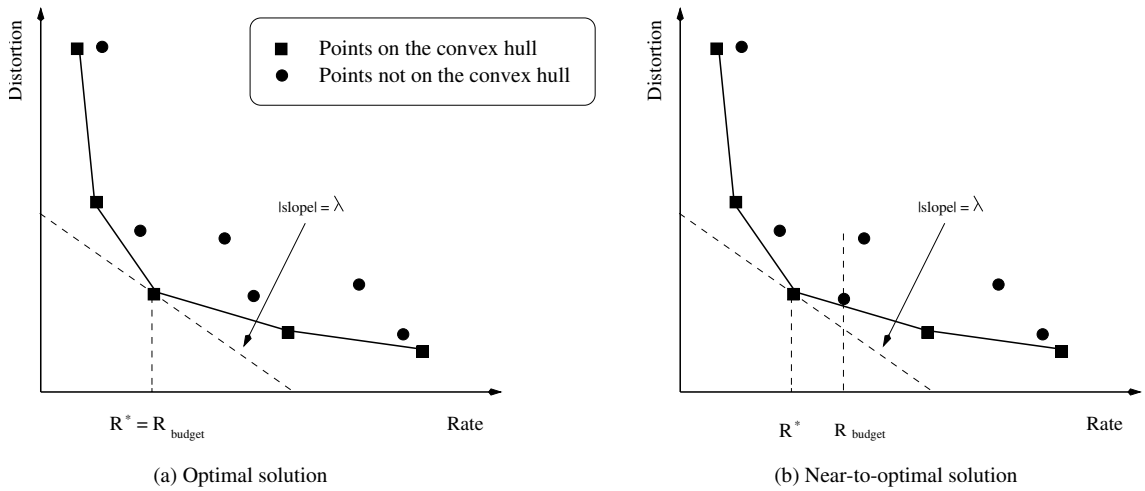

Figure 2.6: Rate-Distortion Curves

pairs. When the optimal solution to the constrained problem lies on the convex hull, solutions to the constrained and to the non-constrained problems are the same (Fig: 2.6 a). Otherwise, the result of the optimization is a point on the convex hull, and the solution is not strictly optimal. The results show, however, that the algorithm is highly efficient and can be considered essentially optimal [124].

This method inspired the work of Ramchandran and Vetterli [105], where the goal was to obtain the optimum wavelet-packet decomposition in a rate-distortion sense jointly with optimum associated quantizers. This was done in a pure intra-frame approach. Also, they stated that their results could be applied to quad-tree segmentation as well. In the work of Reusens [107] this was further developed, and a solution was provided for optimal quad-tree partitioning in a rate-distortion sense together with optimal representation models. In this work, though a motion-compensation coding model was used, the definition of the quad-tree relied in a purely intra-frame technique, so that the problem of temporal coherence of the partition was not addressed.

If the signal blocks are not independent, the problem is far more complicated and the above solutions are not directly applicable. In [104], solutions are proposed for both temporal (different frames in an MPEG GOP) and spatial (pyramid-based multi-resolution coding) dependencies.

An in-depth study of rate-distortion methods and its applications to video coding can be found in [91, 25, 131].

In Chapter 5, a rate-distortion algorithm is used to solve the general problem of finding the best representation for an image sequence into the framework of segmentation-based coding systems. In Chapter 7 a variant of this algorithm is presented that allows dealing with

content-based functionalities in a scalable coding system.

Chapter 3

Methods oriented at coding efficiency

In this chapter, the principal coding systems intended at coding efficiency are analyzed. This analysis is carried out taking into account the features that are most relevant to the work reported in this thesis. That is, the capabilities of the coding techniques to support scalability, the methods used for optimization, and the possibility to handle content-based functionalities.

Note that most video standards do not specify how to control the bit-rate, the exact rate control of the encoder is generally left open to user specification.

For the sake of comparison, two main categories are considered: block-based and segmentation-based coding systems.

3.1 Block-based methods

In block-based coding (H.261/H.263, MPEG-1, MPEG-2), the image is decomposed into fixed-shape blocks and discrete transform methods (cosinus or other) are used to efficiently explore spatial correlations between nearby pixels within the same image. Motion compensation is also performed on these blocks.

These methods have a number of advantages that have determined that they are at the heart of the most used standards and applications:

- Simplicity: As they deal with regular blocks, partitioning of the image is straightforward. In addition, there exist a large number of texture coding techniques aiming at coding blocks of pixels.

- Hardware implementation: All these methods are easy to implement on hardware, and in fact, a variety of VLSI implementations exist.

- Good performance: They provide very good image quality for moderate compression rates. However, annoying blocking artifacts can appear at low bit-rates.

The drawbacks of these systems are that they are not suitable for content-based functionalities and the low visual quality at low bit-rates due to blocking artifacts.

In this section, the most important video coding standards that are based in the block-based approach are reviewed mainly in terms of its support of content-based functionalities, scalability and coding optimality.

3.1.1 H.26x

Recommendation **H.261** [42] describes the video coding and decoding methods for the moving picture component of audiovisual services at the rate of p x 64 kbit/s, where p is in the range 1 to 30. This standard is intended for carrying video over ISDN, in particular for face-to-face videophone applications and for videoconferencing. It uses block motion-compensated Discrete Cosine Transform (DCT) structure for encoding.

The input signal consists of non-interlaced video frames in CIF (352x288 pels) or QCIF (176x144 pels) format. This signal must be in the form of one luminance component (Y) and two chrominance components (C_b, C_r). These chrominance components are down-sampled by a factor of two in both the horizontal and vertical directions.

A four layer hierarchical structure is used, consisting in frame, group of blocks (GOB), macroblock (MB) and block layers. The lowest level is an 8x8 block of pixels. Macroblocks are formed by 4 luminance (Y_1, Y_2, Y_3, Y_4) and two chrominance (C_b, C_r) 8x8 blocks

Motion estimation-compensation is done for each macroblock. Depending on the accuracy of the prediction, a decision on the coding mode (inter/intra/skip) is done. A loop-filter is used in the compensation step to eliminate the high frequency components of the signal. Integer pixel motion estimation is used, with a maximum motion vector displacement of 15 pixels in the horizontal and vertical directions. DCT coding is performed on the 8x8 pixel blocks of the macroblock to remove spatial redundancy.

This standard was designed for coding efficiency, with no support for content-based functionalities. It is either not able to deliver scalable bitstreams.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: Not supported
- **Rate Control**: Rate-control is a simple feedback loop based on "buffer fullness". If the buffer is too full, the quantization scale factor is increased to reduce the data. It is possible to adjust this parameter at the macroblock or at the GOB level, but only a small adjustment with respect to the value of the most recent quantizer is permitted.

- **Content-based functionalities**: Not supported.

As new video applications require compression performance and channel error robustness that can not be achieved by H.261, a new recommendation was proposed. **H.263** [43] was designed for very low bit-rate coding applications. H.263 is based on H.261 but it is significantly optimized for coding at low bit-rates, providing better picture quality with little additional complexity. It can be used for bit-rates lower than 64 kbps.

The basis, as in the H.261 recommendation, is a block motion-compensated Discrete Cosine Transform (DCT) structure for encoding, with higher efficiency than H.261 encoding. The main differences with H.261 are:

Motion estimation accuracy. Half-pixel accuracy motion estimation is used for better accuracy of the prediction.

Loop-filter. H.263 does not use the loop-filter to eliminate the signal high-frequencies because the improvements that provides the half-pixel motion estimation makes it unnecessary.

Optional modes: Includes four optional modes aiming at improving compression performance: unrestricted motion vectors (allows motion vectors to point outside the picture boundaries), advanced prediction (allows for the use of four motion vectors per macroblock), PB frames (the frame structure consists of a P-picture and a B-picture. The P picture is forward predicted from the previously decoded P or I picture. The B picture is bidirectionally predicted from the previously decoded P picture and the P picture currently being decoded) and syntax-based arithmetic coding (replaces the default VLC coding by arithmetic coding, which is more efficient). The first two modes are used to improve inter-picture prediction. The PB-frames improve temporal resolution with little bit-rate increase. The syntax-based arithmetic coding mode can reduce the bit-rate without affecting the decoded picture quality.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: There is no support for scalability in the basic H263 standard.

- **Rate Control**: It is possible to adjust the texture quantization parameter at the macroblock or at the GOB level, but only a small adjustment of ± 1 or ± 2 in the value of the most recent quantizer is permitted.

  As this standard is designed for VLBR, the rate control mechanism allows variable frameskip and thus, it is up to the RC algorithm to make appropriate decisions on both spatial and temporal coding parameters.

- **Content-based functionalities**: H.263 is also designed for coding efficiency and therefore there is no support for content based functionalities.

H.263+ [44, 17] is an extension of the H.263 recommendation providing 12 new negotiable modes and additional features. The objective is to broaden the range of applications and to improve compression efficiency. H.263+ (or H.263 Version 2) is backward compatible with H.263. It allows the use of a wide range of custom source formats. Picture size, aspect ratio and clock frequency can be specified as part of the bitstream. Another important improvement over H.263 is scalability.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: H.263+ provides SNR scalability as well as spatial and temporal scalability. This can improve the delivery of video information in error-prone, packet-lossy or heterogeneous environments by allowing multiple display rates, bit-rates, and resolutions to be available at the decoder.
- **Rate Control**: The optional *Modified Quantization Mode* allows more flexibility in the rate control by permitting the modification of the quantizer of each macroblock to any value (without the ± 1 or ± 2 restrictions in plain H.263).
- **Content-based functionalities**: Not supported.

H.263++ [45] is an extension of H.263+. It considers adding more optional enhancements to H.263+ in order to improve coding efficiency and error resilience. There are no changes in RC, scalability or content-based functionality with respect to H263+.

H.264 [140, 139] is the newest video coding standard of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The main goals of the H.264/AVC standardization effort have been enhanced compression performance and provision of a "network-friendly" video representation addressing "conversational" (video telephony) and "non-conversational" (storage, broadcast, or streaming) applications. H.264/AVC has achieved a significant improvement in rate-distortion efficiency relative to existing standards.

The basic functional elements (prediction, transform, quantization, entropy encoding) are little different from previous standards (MPEG1, MPEG2, MPEG4, H.261, H.263); the important changes in H.264 occur in the details of each functional element. Some highlighted features of the design that enable enhanced coding efficiency include the following enhancements:

- Variable block-size motion compensation with small block sizes
- Quarter-sample-accurate motion compensation
- Motion vectors over picture boundaries
- Multiple reference picture motion compensation

- Small block-size transform
- Hierarchical block transform
- Arithmetic entropy coding
- Context-adaptive entropy coding

Detail on these techniques can be found in the references.

H.264 includes a new structure for video coding development, separating the codec design into two distinct layers: a video coding layer (VCL) to efficiently represent the video content, and a network adaptation layer (NAL) to customize and packetize the content for delivery over a particular network environment.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: There is no support for scalability.
- **Rate Control**: Rate control is not a part of the H.264 Standard, but the standards group has issued non-normative guidance to aid in implementation [26, 27, 130]. It can be considered an evolution of the approach in MPEG-2/TM5 [41] techniques with accommodation for the more general prediction methods of H.264 and providing more flexibility to scale the granularity of control.

  Rate control may be pursued to different levels of granularity - such as picture, slice, macroblock row or any contiguous set of macroblocks. That level is referred to as a "basic unit" [26, 27] at which rate control is resolved, and for which distinct values of the quantization parameter (QP) are calculated.

  To guarantee stability and to minimize perceptible variations in quality. For difficult sequences having rapid changes in complexity, QP-demand may oscillate noticeably, so a rate limiter is applied which typically limits changes in QP to no more than ± 2 units between pictures.

  Any compliant decoder is equipped with a buffer to smooth out variations in the rate and arrival time of incoming data. The corresponding encoder must produce a bitstream that satisfies constraints of the decoder, so a virtual buffer model is used to simulate the fullness of the real decoder buffer.

  H.264 provides 7 modes for inter (temporal) prediction, 9 modes for intra (spatial) prediction of 4x4 blocks, 4 modes for intra prediction of 16 x 16 macroblocks, and one skip mode. Each 16 x 16 macroblock can be broken down in numerous ways. Thus, mode selection for each macroblock is a critical and time-consuming step that enables much of the dramatic bitrate reduction.

Selection of the optimal mode is done by a rate-distortion optimization (RDO) algorithm [138], which essentially involves 1) an exhaustive pre-calculation of all feasible modes to determine the bits and distortion of each; 2) evaluation of a metric that considers both bitrate and distortion; and 3) selection of the mode that minimizes the metric.

RDO is complementary to rate control; these two aspects of the problem are decoupled because a fully coupled optimization would require a more expensive iterative solution.

- **Content-based functionalities**: Not supported.

3.1.2 MPEG-1

The MPEG-1 standard [81, 126] aims at representing moving pictures and associated audio for storage and transmission on digital media with a bit-rate up to 1.5 Mbit/s. It covers many applications from interactive systems on CD-ROM to the delivery of video over telecommunications networks.

The MPEG-1 compression is aimed at removing interpel correlation, including the assumption of simple, correlated translational motion between consecutive frames. An adaptive combination of temporal motion-compensated prediction followed by transform coding of the remaining spatial information is used to achieve high data compression.

Each input frame is partitioned into non-overlapping macroblocks, each one comprising 4 blocks of luminance data and 2 blocks of chrominance data, each with a size of 8x8 pixels. Discrete cosine transform (DCT) coding techniques are used on the 8x8 blocks to efficiently explore spatial correlations between nearby pixels within the same image. The resulting 64 DCT coefficients are uniformly quantized. The DC coefficient is encoded using a differential DC prediction method because there is usually strong correlation between the DC values of adjacent blocks.

When two consecutive frames have similar content, inter-frame differential pulse code modulation (DPCM) is used in the temporal axis, employing temporal prediction (motion-compensated prediction between frames). In this case, the estimation is done on a macroblock basis, using only the luminance components, so that only one motion vector is used for each macroblock.

Three different macroblock coding types provide the possibility to encode macroblock information only if it is necessary; this is, if the contents of the macroblock has changed with respect to the contents of the same macroblock in the previous frame. This feature, called *conditional replenishment* is basic for the efficient coding at low bit-rates.

When processing the frames of a video sequence, the MPEG-1 algorithm encodes the first one in intra-frame mode (I picture), without references to any past or future frames. Successive frames are encoded using inter-frame prediction (P pictures). In this case, only

data from the nearest previously coded I- or P-frame are used for prediction. To further reduce temporal redundancy, a bidirectional prediction is also used. In this case, data from the nearest past and future P-pictures or I-pictures are used as reference for the prediction.

To avoid error propagation as well as to provide access points for random access and *fast forward/fast reverse* (FF/FR) functionalities, I-frames are used at regular intervals. The user at the encoder can combine the picture types (I, P and B) in a video sequence with a high degree of flexibility. To further increase the compression ratio, video can also be sub-sampled in the temporal direction prior to coding.

A limitation of the MPEG-1 standard is that it is designed for non-interlaced input signal.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: MPEG-1 has no support for scalability, though temporal scalability can be implemented by dropping B frames and possibly P frames. It is possible to decode a sequence at a lower temporal resolution by simply skipping B frames. However, because B frames are the most efficiently compressed, only small savings are obtained by omitting them. After the B frames, P frames can also be dropped, again with relatively small savings. This will leave a stream of I-frames only. Chang and Zakhor [13] have implemented MPEG-1 temporal scalability by storing the frames within a Group-of-Pictures (GOP) in a specific order.

- **Rate Control**: The MPEG-1 rate control is done by permitting the selection of different quantizer values for each coded macroblock. This allows a distribution of the bits among the macroblocks depending on the properties of the macroblock block being coded, resulting on an improved quality for a given bit-rate or a lower bit-rate for a fixed quality.

  For instance, in [106], a R-D optimization strategy for MPEG as well as JPEG coders is presented. It is based on an optimal thresholding of the quantized DCT coefficients, that allows to drop the less-significant coefficients in the image or video frame.

- **Content-based functionalities**: Not supported.

3.1.3 MPEG-2

MPEG-2 [133, 126, 81] can be considered an extension of MPEG-1 to provide a solution to applications not covered by MPEG-1. The primary application targeted during the MPEG-2 definition process was the all-digital transmission of broadcast SDTV (Standard Definition TV) quality video at coded bit-rates between 4 and 9 Mbit/sec. In fact, it is a superset of MPEG-1 and an MPEG-2 decoder will be able to decode an MPEG-1 compliant bitstream. However, the MPEG-2 syntax has been found to be efficient for other applications such as those at higher bit-rates and sample rates (e.g. HDTV - High Definition TV).

Perhaps the most significant enhancement over MPEG-1 is the addition of syntax for efficient coding of interlaced video (e.g. 16x8 block size motion compensation, Dual Prime, etc.).

Several other more subtle enhancements (e.g. 10-bit DCT DC precision, non-linear quantization, VLC tables, improved mismatch control) are included which have a noticeable improvement on coding efficiency, even for progressive video. In order to keep implementation complexity low for products not requiring the full video input formats supported by the standard (e.g. SIF to HDTV resolutions), so called "Profiles", describing functionalities, and "Levels", describing resolutions, were introduced to provide separate MPEG-2 conformance levels.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: The scalable extensions permit the division of a continuous video signal into two or more coded bit streams representing the video at different resolutions, picture quality (i.e. SNR), or picture rates.

- **Rate Control**: The most influential coding parameter with regard to picture quality is the quantization parameter (QP) used for texture coding. This parameter can be selected for the entire frame or change from macroblock to macroblock. In most implementations, it is selected based on a measure of buffer fullness so that the target bitrate can be obtained.

  Rate-distortion theory has been applied successfully to optimize the frame type and/or the quantizer selection in MPEG systems. For instance, in [104, 58] the best encoding strategy is selected at the frame level. Another example can be found in [57], where the quantization matrix and the best quantizer for each macroblock are jointly optimized.

- **Content-based functionalities**: Not supported.

3.2 Segmentation-based coding systems

Segmentation-based coding methods (SBCS) [55, 86, 134, 38, 119, 107] rely on an adaptive partition of the image, that can be described as a set of textured regions with arbitrarily shaped contours. Texture and contour information can be adaptively coded using an appropriate representation model for each image segment.

The goal is efficient sequence coding, i.e., maximal quality of the decoded images for a given coding cost (or the dual problem, minimal coding cost for a given image quality). The segmentation has to produce a set of regions (partition) suitable for coding purposes. This can be done by ensuring that the regions are homogeneous in some sense (e.g. gray level,

color or motion). Due to this homogeneity, the information of each region can be separately coded in an efficient manner.

A common problem found in segmentation-based coding systems is the high cost of coding the contour information. Accurate contour representation is very important for the quality of the decoded images because the errors that appear in the decoded images have an important visual effect. This leads to the use of lossless or near-lossless coding techniques. However, a partition of the image with many regions will produce highly homogeneous regions, with an easily codable texture, but the cost of contours will be very high. On the contrary, a partition with few regions will lower the cost of contours but the regions can be rather inhomogeneous. Balancing adequately the coding cost between contours and texture is very important for the performance of the coding system. In this sense, bit-allocation techniques (see sections 2.3 and 5.2) are specially important in SBCS because they can help to select the optimal partition besides the optimal texture coding technique for each resulting region.

3.2.1 Segmentation techniques

The purpose of segmentation is to divide the image into a set of connected regions $\{R_i\}$ so that every pixel in the image is related to one, and only one, region. This set of regions is said to form a partition of the image. Each region in the partition receives a different label [72].

Image segmentation is an ill-posed problem [8]. For a given image, different partitions can be obtained depending on the homogeneity criteria. So, the segmentation approach has to be selected depending on the final specific application where it is to be used. In the framework of segmentation-based video coding the goal is the quality of the final image representation as well as its coding cost. This means that, to obtain a suitable partition for coding purposes, the special characteristics of the coding techniques that are used should be taken into account. Thus the segmentation process not only has to describe the information in the scene, but has to lead to a low cost representation.

When working with video sequences, removing temporal redundancy is a key factor for performance. In this context, a good segmentation has to provide regions with contours and texture that are efficiently predictable by motion compensation. This means that regions have to be correctly related in the temporal domain.

Additionally, for the work in this thesis, the segmentation algorithm must be able to deal with content-based functionalities, such as representation of semantic objects, object tracking and scalability.

Three major steps can be outlined in a segmentation process [115]: simplification, feature extraction and decision. The simplification step aims at reducing the complexity of the process by removing irrelevant information. The feature extraction step involves the choice of the type of homogeneity to be considered (this is, the choice of the feature space) and the process of

estimation of the feature (or features) that will be analyzed looking for homogeneity. This analysis of the data in the feature space is performed later in the decision step, where the position of the boundaries in the decision space is established. These boundaries separate data areas that contain elements sharing the same characteristics in the feature space.

In [115], the naming of the various segmentation techniques is done according to the feature and decision spaces used by each technique. The naming convention is:

$$\left\{ \begin{array}{c} Inter \\ / \\ Intra \end{array} \right\} \left\{ \begin{array}{c} Feature \\ space \end{array} \right\} \left\{ \begin{array}{c} Decision \\ space \end{array} \right\} segmentation$$

Then, a classification is done according to the decision viewpoint, with two main categories: transition based and homogeneity based segmentation techniques. Transition based techniques intend to estimate the position of discontinuities (that are evaluated in the feature space) in the decision space. To do this, a pre-processing step that defines transition zones is followed by a process of reduction of uncertainty in these zones that provides the final region borders.

The second category is formed by techniques that work by looking, in the decision step, for regions where elements are homogeneous with respect to the feature space. In the literature one can find two main types of homogeneity criteria: deterministic and probabilistic. In both cases, a model is given for each region. In deterministic models the decision on to which region belongs each pixel is taken by computing the distance between the actual pixel value and the different model estimations (one per region) for this pixel. In probabilistic schemes, regions are modeled by a random field and the a posteriori likelihood of the data to be a sample of this random field is computed.

For the purposes of this thesis, segmentation techniques will be reviewed in terms of the feature space they use. Three main categories will be distinguished: temporal, spatio-temporal and spatial techniques.

The goal of temporal or motion-based segmentation techniques is to partition the image into regions that have homogeneous motion. The notion of motion homogeneity depends upon the motion model considered. This approach usually results in limited accuracy, specially on motion boundaries [21]. Examples of motion-based techniques can be found in [1] and [39].

In spatial segmentation features are estimated from spatial information (color or gray level). Features such as size, contrast, spectral content or dynamic characterize visually relevant objects. For each feature, a different homogeneity criterion can be applied.

Coding systems that only utilize spatial segmentation face the problem of exploiting temporal redundancy. Although some different ways to solve this problem exist [2, 118, 112, 142, 93, 103], perhaps the most useful method uses a time recursive segmentation of the sequence [95, 75, 64]. In this case the sequence is regarded as a 3D signal and the segmenta-

tion is carried out over a window of several frames that is shifted along the sequence as the processing of the sequence advances along the time axis. In this approach, the partitions of former frames are used to initialize the segmentation procedure of the new frames.

There are also segmentation techniques that directly combine both spatial and temporal homogeneity in the same partition. This spatio-temporal approach seems to provide the most promising results. As there is a strong interaction between the estimation of the moving objects boundaries and their motions, it makes sense to use spatial and motion information in the segmentation step. This is the case of the joint estimation of motion and segmentation. The segmentation is expressed as a relaxation problem based on a Markov Random Field (MRF) modeling and a Bayesian criterion [21]. In this context, the spatio temporal segmentation and the motion are simultaneously estimated [20].

Another example of spatio-temporal segmentation can be found in [117], where spatial (size and contrast) and temporal (motion of regions) features are combined to obtain the final partition.

In the sequel, only spatial and spatio-temporal homogeneity based segmentation methods will be discussed. These methods are better suited for coding purposes because of its superior accuracy on the position of boundaries. In general, spatial and spatio-temporal segmentation techniques use homogeneity-based methods in the decision step. This is because the transition-based approach presents problems when used in spatial segmentation (lack of robustness, lack of accuracy at the transitions). However, some spatial transition-based segmentation techniques have been presented [77, 18]. These methods are mostly intended for video analysis and indexing and are not well suited for coding purposes.

In the homogeneity-based approach, the homogeneity estimation is followed by a partition optimization step, where an initial partition or a set of markers is modified to reach an optimum in the homogeneity criterion sense. Two processes are involved in this modification: split and merge. Regions that do not fulfill the given homogeneity criterion can be split if its division leads to a better configuration. Usually, only a predefined set of possible divisions is taken into account to avoid excessive computational load. A well known example of this is quad-tree split. On the other hand, regions can be merged if they can be efficiently described by a single set of parameters. The process can be initialized by defining a set of markers covering partially or totally the space or by a previous split process, as in *Split and Merge* [142]. The basic approach to merging is region growing. One of the most important region-growing algorithms is the watershed [79, 9]. This mathematical morphology tool detects the minima of the gradient of the gray level image and grows these minima according to the gradient values. The growing of these minima continues until they completely cover the space. The points where the different minima contact in this propagation process are the region borders. Examples of segmentation techniques that use the watershed can be found in [119, 99].

Even though spatio-temporal segmentation provides better result in video coding appli-

cations, purely spatial segmentation techniques are important by several reasons [72]. Any coding technique needs an intra-frame coding mode, and intra-frame segmentations can utilize only spatial information. Moreover, spatial segmentation is useful even in systems that use motion-based segmentation because there are always regions in which the motion approach will fail. This is the case, for instance of new objects appearing in the scene.

The coding system that is presented in this thesis uses a spatio-temporal segmentation based on the watershed algorithm [119, 117]. There are several reasons for this choice: the coding efficiency is ensured because the temporal connectivity of the regions is preserved along the sequence. This allows the use of an inter-mode for the texture and partition coding steps. Additionally, ensuring that the time evolution of the regions is predictable allows the introduction of content-based functionalities. As the segmentation process is based on a multi-resolution approach, with a hierarchy of partitions that describe the scene with various levels of detail, it is very appropriate for scalability purposes. Complete details are given in Chapters 5 and 7.

3.2.2 Coding of partition and texture

Once the segmentation step has been performed, the resulting partition (shape and localization of the regions) and the texture (gray level or color information) of the resulting regions have to be encoded and transmitted. Usually, the process is to encode first the contour and later the texture, in a closed loop approach. This way, errors due to lossy contour encoding are avoided.

Two main differences can be noticed with respect to block-based coding systems. The first one is that in block-based systems there is no need of the partition coding step because it is fixed a priori (rectangular blocks). The second difference is that the arbitrarily-shaped regions that appear in SBCS make possible a better adaption to the local image characteristics but also increase the complexity of the texture coding methods with respect to the block-based techniques.

The regions resulting after a segmentation process are statistically quasi-stationary and should therefore enable higher data compression ratios [34]. However, traditional block-based texture coding techniques can not exploit efficiently this quasi-stationarity because they perform poorly on border blocks. In this case, the overhead imposed by the need to code the contour information, can result in a performance loss.

There are many texture and contour coding techniques that can be used in this kind of systems. Some of them are an evolution of existing block-based techniques, with modifications to deal with blocks located at region borders, while others are designed specially for SBCS. We can cite for example, Shape-Adaptive DCT [127, 46], Shape-Adaptive Wavelet [50] and padding methods [51] as examples of the first category, while the generalized orthogonal transform method in [35] is on the second category.

Due to non-stationarity behavior of images, some texture coding techniques may be appropriate in some regions, whereas they may be particularly inefficient in some others. Therefore, the approach followed in the work in this thesis is to code each region with different techniques and then use an optimization algorithm to find the best one in each case. All the above commented techniques are adequate because they can be used in a scalable coding and the trade-off between coding cost and quality can be chosen for rate-control purposes.

The actual techniques that are being used in the coding system that is developed in this thesis will be further commented in Chapter 5.

Partition coding is necessary on SBCS to inform the receiver of the shape (contour) and spatial/temporal localization of the regions. Contour coding can be lossless or lossy. Because the human visual system is very sensitive to errors in the localization of the contours of the regions, lossless or near-lossless approaches are popular. However, these methods are expensive.

In the literature it is possible to find a wide range of contour coding techniques. Among the lossless techniques, can be pointed out Chain Code (CC) [30], Morphological Skeleton [122, 66], transition points [101] and Quadtree decomposition (a particular case of Skeleton [53]). Multigrid Chain Code (MGCC) [80, 32] is an example of a near-lossless technique (it is an extension of Chain Code). Finally, some examples of lossless techniques are Fourier descriptors [143, 92], and polygonal [23] and Spline [121] approximations.

Note that partition coding has to deal not only with the contour of the regions, but also with the temporal evolution along the video sequence. This is important because it will be the basis for an efficient inter-frame mode, thus helping to effectively remove the temporal partition redundancy. Additionally, in Chapter 4, we will see that this feature is essential to enable content-based functionalities.

The reader is referred to [134] for an in-depth analysis of these subjects.

For the work in this thesis, the contour coding methods that have been selected are Chain Code and Multigrid Chain Code. The basic reasons for this choice are quality, coding efficiency and its adequacy for scalability applications. Further details will be given in Chapter 5 and 7.

3.3 Examples of coding efficiency oriented SBCSs

Currently, there are no standards defined that can be inscribed in this category. However, there are many coding techniques, and some more or less complete coding systems. In this section, some examples will be given of these kind of coding systems, with special attention to the ones with a formulation that is close to the one used in the work presented in this thesis.

3.3.1 Joint optimization of representation model and frame segmentation

The technique presented in [107] relies on the joint optimization of two degrees of freedom, namely adaptive representation model and adaptive frame support partition. Optimization is performed in a rate-distortion sense. In a first step, the algorithm build up basements of a set of possible coding scenarios. The current frame is recursively partitioned into variable size cells along a quadtree structure. Each cell of the tree is approximated within a given set of available models. For each model, approximation error and encoding parameters corresponding to each individual cell are recorded. The set of solutions therefore comprises all possible partitions following a quadtree structure together with all possible coding models associated to each cell. Each solution results in different rate and distortion values. In a second stage, the solution leading to the global optimum in a rate-distortion sense is selected and encoded. The algorithm determines the strategy which minimizes the distortion for a given rate budget. The technique was optimized for low bitrate applications.

One drawback of this technique is that, though a motion-compensation coding model was used, the definition of the quadtree relied in a purely intra-frame technique, so that the problem of temporal coherence of the segmentation was not addressed.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: Not supported.
- **Rate Control**: Rate-distortion Lagrangian algorithm.
- **Content-based functionalities**: Not supported.

3.3.2 Segmentation based on the MDL principle

The Minimum Description Length (MDL) criterion [108] is aimed at minimizing the global representation cost (or description length) of a chain

This formalism can be applied to find the appropriate representation model in the coding of still images or video sequences. Generally speaking, a segmentation-based video coding system can be modeled with three sets of parameters: The segmentation S_i, the motion parameters θ and the prediction error image E.

This is, given these three sets of parameters the decoded image is fully characterized. The overall coding cost for each image is the sum of the costs of these three parts.

In [98, 89] this criterion is applied to a segmentation-based coding system and used to obtain a segmentation of the image with regions that are uniform on a motion sense, with a minimum global coding cost. To do this, an initial quad-tree segmentation is performed on the image. Then, a multi-level relaxation is done on the obtained blocks, merging the resulting blocks in order to minimize the MDL criterion. Once this spatial segmentation is obtained,

a spatio-temporal segmentation is performed. The relaxation-merging step is repeated, this time on regions that are uniform on a motion sense. The goal is again a minimization of the MDL criterion.

Region merging is done by taking a region and trying to merge it with a neighbor region. The coding cost of the merging is compared with the sum of the costs of the original regions. If the cost of coding the original regions is higher than the result of the merging, the merging is accepted. Otherwise, the merging is attempted with another neighbor region. This is done for all the regions.

This system is designed only for coding efficiency and it is not easy to extend it to deal with content-based functionalities.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: Not supported.
- **Rate Control**: Minimization of coding cost based on the MDL formalism.
- **Content-based functionalities**: Not supported.

3.3.3 SESAME: SEgmentation-based coding System Allowing Manipulation of objEcts

SESAME [117, 16] is a region-based generic video coding system. The coding strategy relies on a joint optimization in the Rate-Distortion sense of the partition definition and of the coding techniques to be used in each region. This optimization creates the link between the analysis and the synthesis part of the coder. The analysis defines the time evolution of the partition, as well as the elimination or the introduction of regions that are homogeneous either spatially or in motion. The coding of the texture as well as of the partition relies on region-based motion compensation techniques.

The *Projection* block applies motion information to the partition of the previously coded frame #(n-1), to obtain a projected partition for the current frame to be coded #(n). The resulting partition is the basis for the current image segmentation. The projected partition provides the time evolution of the regions in the previous partition, i.e., it tracks the regions over each frame of the sequence.

For the intra-frame mode, the *Projection* step is not related to the previously coded partition. In this case, an initial partition with a predefined number of regions is constructed.

The *Partition Tree* block constructs a set of partition proposals from the projected partition, as shown in Figure 6.4. These partition proposals define a reduced set of regions which are candidate to belong to the final partition. The various levels of segmentation represent the image with various levels of detail, so fluctuations with respect to the projected partition

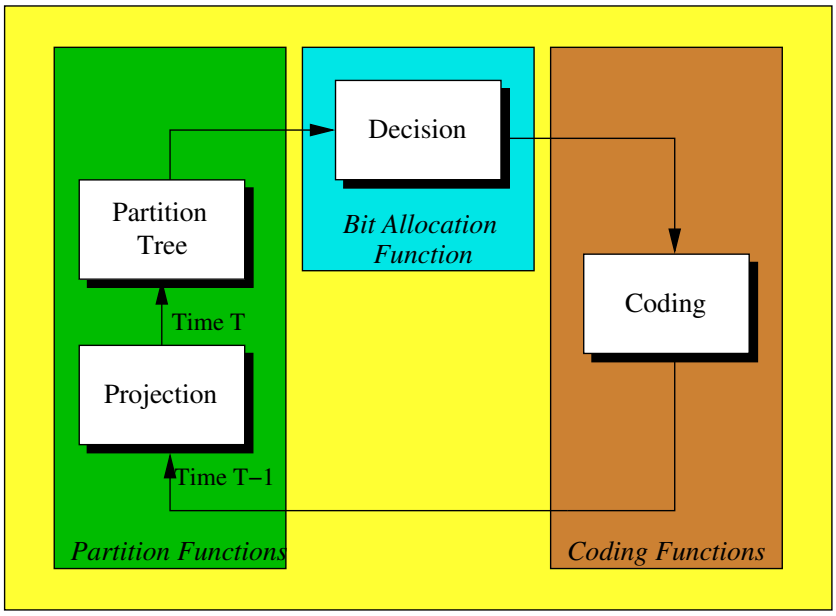

Figure 3.1: Scheme of the segmentation-based coder

can be properly introduced by selecting regions from the different partitions. As the partition proposals are derived from the projected partition, the temporal coherence between the partitions of successive frames is preserved.

On the one hand, a reduced set is necessary in order to limit the computational complexity of the decision that will be taken afterward. On the other hand, the partition proposals should be carefully created to allow an efficient and pertinent decision. The final partition will in fact be composed of regions issued from the different levels of the Partition Tree.

The *Decision* block takes the proposals from the Partition Tree and makes a decision on which regions will belong to the final partition, and which coding technique will be used for every region. This step is based on a rate-distortion based optimization algorithm.

Finally, the *Coding* block takes the results of the decision block and codes the image. In the SESAME description, this block is divided into four sub-blocks since four different types of information have to be coded: the motion parameters, the texture parameters, the contours of the partition and the information related to the decision.

The strong side of this coding system is that it is based on a hierarchical representation of the image content, with a partition that is constructed with regions at different resolution levels. Moreover, to enhance coding efficiency, the system can follow the evolution of regions along the sequence. These facts introduce a slight overhead that can affect a little the system in terms of coding efficiency, but that make it very appropriated for content-based applications (See Section 4.1.3).

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: There was no support for scalability in the basic SESAME algorithm. In this work, it is shown how scalability ca be introduced in an optimal way in this coding system (See Chapter 5).
- **Rate Control**: Basically, three types of information are relevant: texture parameters, shape parameters and motion parameters. Bit allocation is done with a rate-distortion based Lagrangian algorithm. The overall bit budget for every frame was set in a simple feedback loop based on "buffer fullness".
- **Content-based functionalities**: The possibility of addressing content-based functionalities was already outlined in [117]. In [63], a modification of the basic SESAME coding scheme that allows content-based selective coding is presented. Some of the possible content-based capabilities of the algorithm have been developed in the framework of this thesis and are presented in Chapter 7.

Complete details about SESAME are provided in Chapter 6.

3.4 Final comment

All the coding systems and standards presented in this chapter were designed for coding efficiency. For this reason, most of them are not able to deal with the new functionalities that are required by the more sophisticated services that are being developed. In the last years, many efforts have been devoted to develop new algorithms, standards and coding systems to cope with these requirements. In Chapter 4, a brief presentation of content-based coding systems will be given.

Chapter 4

Content-based coding

The generalization of multimedia communications and the availability of new resources and technique originated the demand for more powerful and complex applications, with higher levels of efficiency and end-user interactivity. Examples of these new demands are content-based scalability, content-based indexing and browsing systems as well as improved edition and composition functionalities.

The existing multimedia standards, H26x, MPEG-1 and MPEG-2 are not flexible enough to address the requirements of such a diverse set of applications. In this context, the necessity of new and more powerful representations have arise. For this reason many efforts are being devoted to develop content-based techniques.

In content-based coding the main goal is to represent the input signal in a way that allows the manipulation of the components of the scene. In this case the primitives subject to coding are not pixels, blocks of pixels or regions formed with some uniformity criterion, but higher level entities, with a semantic content, named in the MPEG-4 terminology video objects (VO). This new approach will allow a kind of interaction more natural for human beings because the representation is close to human perception.

The definition of the Video Objects is a critical part in this process. Many segmentation algorithms have been developed, most of them aimed at detecting regions that are homogeneous in some sense. For a content-based representation these segmentation techniques are not sufficient because they cannot catch the required semantic component that is essential to content-based systems. In some systems this can be solved by using a supervised approach [12, 62], where the end-user can select and group homogeneous regions given by a previous "classical" segmentation. In [62] this is done in the first frame of the sequence and then the system can track the selected objects along the sequence automatically using an appropriate tracking algorithm [70].

Other methods exist that use an unsupervised approach to define Video Objects. Usually, these methods are application dependent or designed for a particular type of object (for

instance, faces or head and shoulder). This is currently a very active research area, where much work is to be done before general and robust methods are available.

Nowadays, there is only one established standard for content-based applications, MPEG-4 and the MPEG committee is working in the definition of a new standard, MPEG-7 [4] that is further based on the content-based paradigm. In MPEG-7, a description of the audio-visual data (meta-data) is provided in addition to the data itself. This makes possible to get information about the audio-visual data without the need of performing the actual decoding of these data. Moreover, the standard also allows data to be conveyed back from the terminal to the transmitter. All these features will provide a higher level of content-based interactivity.

4.1 Examples of content-based coding systems

4.1.1 Model-based coding

The essence of model-based coding is to find an appropriate model that allows the representation of image contents through a set of parameters. Then, reconstructions from these parameters can be achieved. The advantages of this approach are a potential high compression ratio, and geometric distortions that are more natural to the human eye than the introduced by traditional block-based methods.

An early example of model-based coding was SIMOC. Aiming at an improved image quality for low bit-rate video transmission, the SIMOC (SImulation Model for Object-based Coding) algorithm followed the approach of object-based analysis-synthesis coding [86, 33].

The underlying source model was that of "planar flexible objects which move translationally in the image plane". The source model leads to a segmentation of images based on detecting changed regions in the image. These regions are then split into moving objects and background being uncovered. A further region type is that of areas which do not comply with the model, model failure regions. The region shapes are then approximated and transmitted to the decoder with any motion and color information necessary to synthesize the entire image.

In a first step the input image is analyzed, i.e. the shape and motion parameters of 2D objects in the image are estimated. An automatic segmentation of the input images is performed in order to estimate the 2D-shape of moving objects in the scene. The 2-D contours of these objects are coded by polygon-approximation. Then, motion vectors, vertex contours coordinates and texture parameters are coded and transmitted to the receiver. At the receiver side, the parameters are decoded and used to reconstruct the image by image synthesis. At the encoder side, the parameters are decoded and the image is reconstructed for analysis of the next input image; additionally, the decoded parameters are stored in a memory.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: Not supported.
- **Rate Control**: Shape, motion an color (texture) parameters can be adjusted to meet the final bit-rate.
- **Content-based functionalities**: Only partially covered because objects are roughly estimated.

4.1.2 MPEG-4

MPEG-4 [125, 48] is a standard designed to provide a generic technology for multimedia communications applications and services [14]. It combines some of the typical features of other MPEG standards with new ones in order to meet the challenges of existing or anticipated manifestations of multimedia. Perhaps the most relevant (from the point of view of the work in this thesis) new feature is interactivity with content. Instead of coding the scene as a whole, in the MPEG-4 architecture one or more visual objects of arbitrary shape are transmitted. An object layered bitstream is used to assist this functionality. At the decoder the end user can interact with the presentation of these visual objects.

The main requirements to be fulfilled by MPEG-4 are [126]:

- Content-based Interactivity.
- Universal accessibility and robustness in error prone environments.
- Coding of natural and synthetic data.
- Compression efficiency.

Bit-rates targeted for the MPEG-4 video standard are between 5-64 kbits/s for mobile or PSTN video applications and up to 4 Mbits/s for TV/film applications.

To enable the envisioned content-based functionalities, various concepts are introduced. A video object plane (VOP) is a region of arbitrary shape that represents a physical object or content of interest present in a frame of a video sequence. In contrast to the video source format used for the MPEG-1 and MPEG-2 standards, the video input to be coded in MPEG-4 is no longer considered a rectangular region. To define VOPs, each frame of the sequence is segmented into a number of image regions. The reunion of successive VOPs belonging to the same physical object is called video object (VO). This is, a VO is a 3D (2D + time) view of an object. All the data belonging to the same VO is encoded and transmitted separately by means of what is called a video object layer (VOL). VO and VOP correspond to entities in the bitstream that a user can access and manipulate (e.g. with cut and paste operations).

The coding of the information of each VOP is block based, in a way similar to the MPEG baseline coders. After encoding the VOP shape information, it is partitioned into

non-overlapping macroblocks identical to those found in MPEG-1 and MPEG-2. An hybrid DPCM/transform encoding process is used. To deal with the incomplete macroblocks that result from the VOP's arbitrary shape, a padding technique is used to fill the macroblock content outside the VOP. This is necessary as the motion estimation and compensation techniques, as well as the texture coding techniques work on square blocks. Support for forward-predicted (P) and backward-predicted (B) VOPs is provided.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: Scalability is provided at VOP level, in contrast to the frame-based approach introduced for MPEG-2. This allows the access or transmission of arbitrarily-shaped VOPs at various spatial or temporal resolutions, thus allowing prioritized transmission, video database browsing, multi-resolution playback of video content or other applications.

- **Rate Control**: It has to deal with the different scalability types and with multiple Video Objects. The rate control has a great deal of flexibility since each object may be encoded at a different frame rate. Also, additional coding parameters are introduced to control the amount of bits used to specify the shape of an object. The distribution of the target bit-rate among the different VO can be based on the size, the motion and the variance of each object.

  More information and examples of rate-control methods can be found in [47, 137, 110, 56].

- **Content-based functionalities**: Different objects in the scene can be coded separately through the use of Video Objects. At the decoder, the end user can interact with video content by accessing and manipulating these objects.

4.1.3 SESAME

Although designed with coding efficiency in mind, the SESAME algorithm presented in 3.3.3 has proved to be easily extensible to support content-based functionalities. The region-based representation model leads to an immediate extension to an object-based model by defining the Video Objects as a group of one or more regions. The region-tracking capabilities already present in SESAME are also a key factor for content-based functionalities. This object-based orientation can be easily reached mainly by introducing some restrictions in the region-merging steps. Also, as the system is very modular, the segmentation process can be adapted to produce partitions with a special focus put on objects of interest in the scene (e.g. moving objects, objects with a given shape, size or color, etc... and combinations of these).

The algorithm offers a good compromise between the ability to track and to manipulate objects and the coding efficiency.

The key difference with MPEG-4 is that SESAME is a segmentation-based coding system while MPEG-4 evolves from a block-based system with some modifications to deal with arbitrary shaped objects. In SESAME the image is partitioned into regions while MPEG-4 only selected objects are coded separately. In this way, the structure of SESAME is well adapted for the *definition* of video objects, a point that is out of the MPEG-4 standard.

Comment on scalability, rate control and content-based functionalities:

- **Scalability**: There was no support for scalability in the basic SESAME algorithm. In this work, it is shown how scalability ca be introduced in an optimal way in this coding system (See Chapter 5).
- **Rate Control**: Basically, three types of information are relevant: texture parameters, shape parameters and motion parameters. Bit allocation is done with a rate-distortion based Lagrangian algorithm. The overall bit budget for every frame was set in a simple feedback loop based on "buffer fullness".
- **Content-based functionalities**: Some of the possible content-based capabilities of the algorithm have been developed in the framework of this thesis and are presented in Chapter 7.

Complete details about SESAME are provided in Chapter 6.

4.1.4 Final comment

Currently, the only established standard supporting content-based functionalities is MPEG-4. It can be seen as an evolution of previous MPEG block-based coding systems, with support for encoding regions of arbitrary shape. On the other side, there are some segmentation-based coding systems that, while originally aimed at coding efficiency, can also address content-based functionalities. In this work, we study various problems that appear in such systems, and we propose solutions to these problems.