Deliverable D1c

Inventory of MPEG-4 Codecs

Public

Information Society Technologies

Project Number	:	IST-004042
Project Title	:	BETSY
Deliverable Type	:	Report

Deliverable Number	:	D1c
Title of Deliverable	:	Inventory of MPEG-4 Codecs
Nature of Deliverable	:	Report, Public
Internal Document Number	:	betsy-imec-25032005d1c-v1_0.doc
Contractual Delivery Date	:	28 February 2005
Actual Delivery Date	:	25 March 2005
Contributing WPs	:	WP1
Author(s)	:	Carolina Blanch (IMEC) Harmke de Groot, Peter van der Stok (Philips)

Abstract

Video communication is rapidly becoming one of the most important parts of the information infrastructure. Unlike text or image, video communication requires huge volumes of data being transmitted in a timely manner, so highly efficient compression must be used. In order to allow the interrelation among different kinds of terminals and networks to be implemented, a big effort was done in the last decade to standardize the video coding process, resulting in a large number of standards such as: H.261, H.263 and H.263+ from ITU; MPEG-1, MPEG-2 and MPEG-4 from ISO and the H.264/AVC. In the BETSY project several video scenarios are presented. According to the scenario characteristics and needs, suitable video codecs need to be selected. This document provides an inventory of MPEG-4 codecs and evaluates the suitability of the different codecs and profiles for our scenarios. The MPEG-4 Part 2 Simple Profile codec appears as a suitable candidate as it provides the needed adaptability to varying bandwidth and processing constraints, while at the same time presenting a good tradeoff in terms of performance and complexity. If the need for higher coding efficiency and resilience capacities justifies a complexity increase, a simple configuration of the Advanced Video Codec (MPEG-4 Part 10) may be taken under consideration.

Keyword list

MPEG-4, video coding, error resilience, scalability, bandwidth constraints

1 Introduction

The purpose of this document is to provide an inventory of standard video codecs and based on its functional and non-functional characteristics to propose those video codecs that are suitable for the video scenarios considered in the BETSY project. First the MPEG standard is introduced and the basics of video coding are explained. In chapter 3 the different profiles of the MPEG-4 Part 2 and Part 10 codecs are presented. The selection of the MPEG-4 Part 2 Simple Profile is justified based on the codecs and scenarios characteristics. Chapter 5 shows the functional attributes of the Simple Profile in terms of Rate – Distortion performance as well as the possible complexity and bandwidth tradeoffs in this codec. Additionally, we list the codec error resilience tools as these will play an important role on its inmunitity to varying environmental conditions causing errors and losses. Chapter 6 explores the non-functional attributes of MPEG-4 codecs explaining the impact of network variations on the video quality. Video adaptation to varying environmental conditions is then proposed as a mean to overcome the video degradation. Finally the conclusion is presented.

2 MPEG standards

Video communication is rapidly becoming one of the most important parts of the information infrastructure. Unlike text or image, video communication requires huge volumes of data being transmitted in a timely manner, so highly efficient compression must be used. In order to allow the interrelation among different kinds of terminals and networks to be implemented, a big effort was done in the last decade to standardize the video coding process, resulting in a large number of standards such as: H.261, H.263 and H.263+ from the ITU VCEG (Video Coding Expert Group), or MPEG-1, MPEG-2 and MPEG-4 from the ISO MPEG group and the H.264/AVC from the Joint Video Team (JVT) where VCEG and MPEG members work together. In Figure 2.1 the international video standardization hierarchy is shown. These standards allow the industry to make major investments with confidence in new products and applications and users to experience easy consumption and exchange of content.

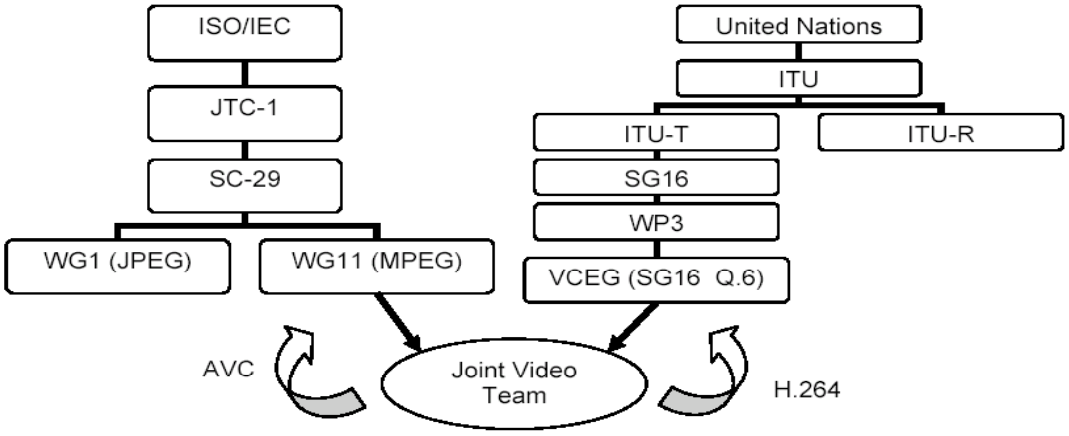

Figure 2.1: International organizations working in video standardization

In 1988, in response to the growing need for a common format for coding and storing digital video, ISO/IEC established the Moving Picture Experts Group (MPEG) with the responsibility of creating, maintaining and updating international standards for compression, decompression, processing, and coded representation of moving pictures, audio and their combination.

In the past decade, MPEG technologies have fuelled the transition from analogue to digital media delivery worldwide by providing the basic foundation for today's mass-market digital distribution platforms. The extraordinary adoption of Digital Versatile Disk (DVD) technology, which is based on a form of MPEG-2 coding, over the last few years is evidence of the impact MPEG technology has had and will have on home entertainment.

All MPEG standards are generic; that is, application independent. They do not specify the operations of the encoder, but just define the syntax of the coded bit stream and the decoding process. Therefore vendors have enough flexibility in the specifications to include specific optimisation elements.
The standards are organized in several Parts (systems, video, audio, conformance testing...), each one with multiple editions, amendments, and corrigenda.

MPEG has been responsible for a series of important standards:

• **MPEG-1**: Standard for coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s. MPEG-1 has been a very successful standard. It was the de-facto form of storing moving pictures and audio on the World Wide Web and was used in millions of Video CDs.

• **MPEG-2**: Standard that allows greater input-format flexibility, higher data rates, and better error resilience to be achieved. Designed when high-definition digital television emerged as a needed, MPEG-2 is ubiquitous at the present time in digital TV broadcasting and DVD-Video and MPEG Layer 3 audio coding (´MP3´) has become a highly popular mechanism for storing and sharing music.

• **MPEG-4** (officially designated as ISO/IEC 14496, in 10 parts, 16 nowadays): It provides the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the next three fields:

- Digital television.
- Interactive graphics applications (synthetic content)
- Interactive multimedia (World Wide Web, distribution of and access to content)

The apparent convergence of the three traditionally separate application areas was evident by seeing how the functional characteristics of each area increasingly emerged in the others.

Around the time at which MPEG-4 Visual was finalised (1998/99), the ITU-T study group began evaluating proposals for a new video coding initiative entitled H.26L. The developing H.26L test model was adopted as the basis for the proposed Part 10 of MPEG-4 in 2001. The new “H.264/MPEG-4 Part 10” standard, entitled “Advanced Video Coding (AVC)”, is being developed by the Joint Video Team (JVT) which consists of members of ISO/IEC MPEG and ITU-T VCEG.

MPEG-7: MPEG initiated in 1996 another standardization project trying to solve the problem of describing audiovisual content to allow the quick and efficient searching, processing, and filtering of various types of multimedia material. Called the Multimedia Content Description Interface or MPEG-7, it was finalised in 2001.
It will be used for indexing, cataloguing, advanced search tools, program selection, smart reasoning about content and more. The standard comprises syntax and semantics of multimedia descriptors and descriptor schemes. MPEG-7 is an important standard because it allows the management, search and retrieval of growing amounts of content locally stored, on-line and in broadcasts.

• **MPEG-21**: Multimedia Framework standard, its aim is to understand how the various elements building the infrastructure for the deployment of applications using multimedia content fit together, and to discuss if there are missing standard specifications for some of these elements.
MPEG-21 includes a universal declaration of multimedia content, a language facilitating the dynamic adaptation of content to delivery network and consumption devices, and various tools for making Digital Rights Management more interoperable. It is about managing and accessing content

The next chapters focus on MPEG-4, as this part of the standard specifically targets the use of multimedia for the fixed and mobile web.

3 Video Coding Basics

Video signal is composed of a sequence of successive still images (frames). The illusion of motion, i.e. the liveliness, arises from the high enough repetition rate for the frames (i.e. frame rate). The sufficiency of the frame rate depends on the image sequence content, especially the amount and speed of motion, typical to the application. Frame rate is therefore an important factor for the quality of an image sequence.

Naturally, video quality is affected also considerably by the resolution of each individual still image, i.e. the number of image pixels used to present it. The resolution of a separate studio quality (recommendation ITU-R.601) TV-picture is 720x576 pixels (pel) and the frame rate is 25 or 30Hz. Respectively, transferring of digital video signal requires very much channel capacity (more than 200Mbps).

Real-time transmission and processing of digital video signal in broadcast (i.e. TV) quality is technically demanding and expensive. Efficient compression of video information is needed in order to use cheap low bit rate connections or store live video digitally. Compression is made in the expense of quality. Lower quality, i.e. smaller resolution, frame rate, and unprecise representation of image pixels is, however, enough for many purposes.

In compression, unnecessary repetition of image information, i.e. redundancy, is tried to remove from the video signal. The information, which is actually needed to understand the image, is maintained as well as possible. In order to achieve high compression efficiency, the similarity, i.e. correlation of neighboring image pixels or blocks is utilized within as large area as possible, e.g. by using a big block size in the compression. This is, however, difficult and increases the complexity and time required for the coding.

In order to achieve high compression efficiency, image information is not totally preserved, i.e. coding causes different kinds of errors, which are typical to the used methods. At low bit rates, coding artifacts substantially degrade the perceived quality. Typical coding artifacts are e.g. blockiness and loss of image details (blurriness), which to certain extent correspond to lowering image resolution.

In principle, video information consists of sequential presentation of colour values (luminance and colour difference values) for image pixels in successive images frames. In order to achieve compression, it is generally beneficial to change the signal space, i.e. the way the signal is represented. Usually it is e.g. advantageous to code the differences of temporally or spatially successive signal values, i.e. signal differences.

Another general principle in increasing coding efficiency is to predict the future signal values, motion, or other useful information on the basis of previously coded values. Since the previous values are at hand both at the transmitting and receiving ends, it is not necessary to explicitly code and transmit the prediction value. As with signal differences above, also differences between actual values and their predictions are usually smaller then the values itself and may therefore be represented with less information bits.

In order to reduce the amount of information required to represent the image, the accuracy of the signal values (signal differences, Discrete Cosine Transform (DCT) coefficients, motion vectors, etc.) may be decreased without any big or even noticeable distortions in the coded image. This procedure is called quantization.

The change of signal space (e.g. as explained before, by forming the difference with the previous or predicted value) and quantization are generally the most important elements of a video compression method.

3.1 Basics of Coding Methods

There are very many types of coding methods for video signals. However, they are usually a compilation of common elementary methods, i.e. they are hybrids (hybrid coding methods). In the following, the principles of few generally used elementary or sub-methods are described. For simplicity, sub-methods are here described as they are used in a traditional block based method, where the input image is divided into a number of equally sized blocks (typically 8x8 pixels) before the coding.

3.1.1 Coding of consecutive images (INTER coding)

In video signal, the most important type of redundancy comes from the repetition of almost identical successive images. The most efficient compression is therefore achieved by utilizing the correlations or similarities between successive images. This kind of coding is generally called temporal or INTER coding.

The most simple way of utilizing the general similarity of successive images is, already before the coding, to leave out some of the frames in the original image sequence. When the sequence contains a lot of motion, it is often necessary, especially at low bit rates, to reduce the frame rate also during the coding. This procedure is generally called frame skipping.

As before, part of the information may be left without coding by the change detection. In this procedure (also being called conditional replenishment), only remarkably changed areas in the previous image are updated or refined. Depending on the amount of motion, the percentage of these areas may be e.g. 10-30% of the whole image area. The whole image is thus not skipped but the coding is made only for a certain number of image blocks. An image block is considered to be changed if the difference between the block to be coded and its temporally aligned predecessor in the previous image differ more than a chosen threshold. The procedure is extremely fast, efficient and simple.

A more advanced INTER coding method is the use of so called motion estimation and compensation. Typically, this means a block based procedure where an area is searched within the previous image (as shown in Figure 3.1), which corresponds to the one being processed. In principle, this area may locate anywhere in the previous image.

The search of corresponding blocks is a tedious operation, which is why the search is typically limited inside a relatively small area (called search area) around the origin of the block being coded.

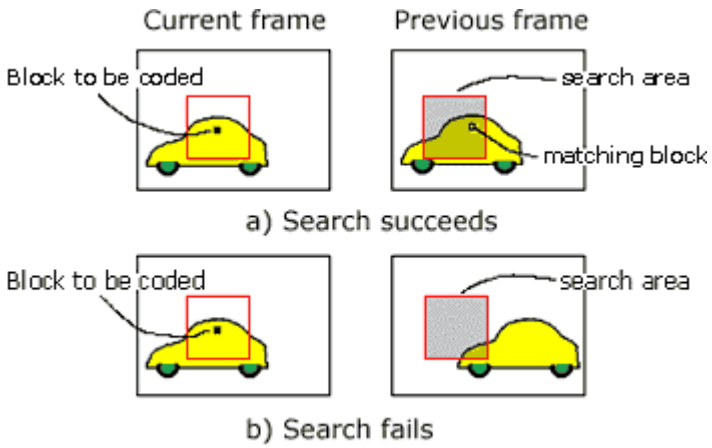

Figure 3.1: Principle of Motion Estimation (ME)

A dualistic procedure for motion estimation and compensation is the prediction of stationary image areas. In practice, the so-called background prediction (Figure 3.2) is performed by updating a similar background (or reference) memory both at the encoder and the decoder, which contains (according to the chosen accuracy or criteria) stationary image areas inside the image sequence. For example, the revealing areas behind a moving object may then be coded simply by making a reference to this memory.

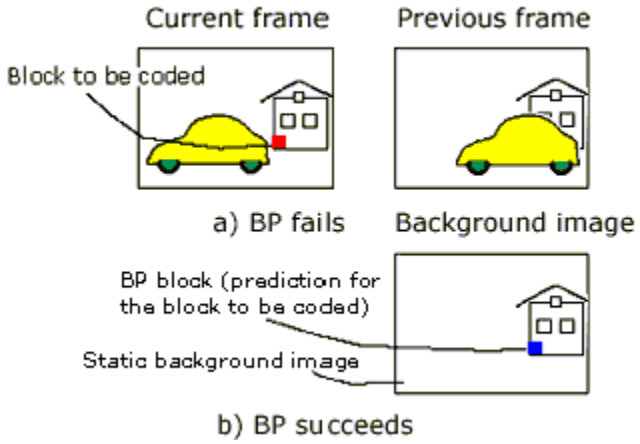

Figure 3.2: Principle of Background Prediction (BP)

3.1.2 Coding inside an image (INTRA coding)

The utilization of the similarities inside an image (the spatial correlations) is made by INTRA coding. In this procedure, almost identical successive image pixels or patterns of pixels are described jointly e.g. by their average or by a chosen symbol (output code).

The most common method for INTRA coding is Discrete Cosine Transform (DCT). In DCT, the block to be coded is transformed mathematically into a transform (frequency) domain, where different kinds of quantization and coding operations may effectively be made. Fast methods for calculating the transformation have been developed, but along motion estimation it is still one of the most laborous operations in most video coding methods and standards.

3.1.3 Entropy coding

The quantized and coded (symbolic) image information, produced by different sub-methods, still contains some residual redundancy, which is removed by special entropy coding methods. These methods utilize the statistical properties of coded information. generally used method is the variable (word) length coding (VLC), in which new shorter code words are allocated for frequent symbols and respectively longer codeword for more rare symbols. Another commonly used entropy coding method is the s.c. run-length-coding (RLC) where a sequence of equal successive code words or symbols are replaced more shortly by a (possibly new) symbol and the number of its repetitions. Entropy coding methods are loss less, i.e. they do not add distortion to the image.

3.1.4 Other components of coding

In practical coding methods, in addition to the elementary methods for compressing video information, mechanisms are needed to make a choice between different coding modes/options, and for controlling quality, buffer fullness (output bit stream), and frame rate. In addition to these, also error protection is needed typically both inside video (source) coding and transmission. In order to achieve good coding results, many elements have therefore to be successfully implemented and optimized.

3.1.5 Standardization of coding methods

Standards for video coding are mainly developed in ITU-T and ISO/IEC. In the former, e.g. videophone standards H.320, H.324, and H.323 have been developed. They are used with digital and analog telephone connections and with LAN-connections, respectively. ITU-T is also developing video coding standard H.263+, with the particular promise of achieving greater coding efficiency.

In ISO/IEC, the commonly used JPEG and JPEG2000 standards for still image compression has been developed, as well as several MPEG standards mainly for storage and retrieval of video signals at different bit rates. Most recent of MPEG standards is MPEG-4, which was developed particularly for Internet applications. Among other unique features, MPEG-4 supports the combining of synthetic and natural audio/visual material, as well as user interaction with the audio/visual content.

3.2 MPEG-4 Video Coding

The MPEG-4 video codec belongs to the class of lossy hybrid video compression algorithms. Figure 3.3. gives a high-level view of the encoder. A new frame arriving to the codec is divided in macro blocks, containing 6 blocks of 8x8 pixels: 4 luminance and 2 chrominance blocks.

The Motion Estimation (ME) [1] enables to exploit the temporal redundancy by searching for the best match for each new input block in the previously reconstructed frame. The motion vectors define this relative position. The remaining error information after Motion Compensation (MC) [1] is decorrelated spatially using a Discrete Cosine Transform (DCT) and is then Quantized (Q) [1] . The inverse operations Q^{-1} and Inverse Discrete Cosine Transform (IDCT) (completing the texture coding chain) and the motion compensation reconstructs the frame as done at the decoder side. Finally, the motion vectors and quantized DCT coefficient are variable length encoded [2] and completed with video header information; they are structured in packets in the output buffer. A rate control algorithm sets the quantization degree to achieve a specified average bit rate and to avoid over or under flow of this buffer.

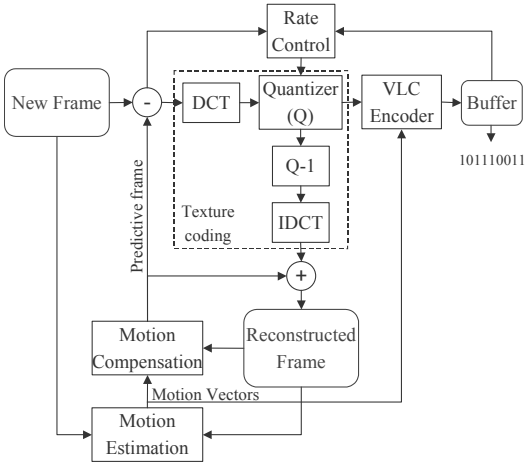

Figure 3.3. Functional view of a hybrid video encoder.

Every input frame (also called Video Object Plane (VOP)) can be coded in three different ways (Figure. 3.4):

1. Intra Frame: a frame may be encoded independently of any other frame. In this case, the encoded frame is called an Intra frame (I-frame);
   A coded I-VOP consists of a VOP header, optional video packet headers and coded macro blocks. Each macro block is coded with a header (defining the macro block type, identifying which blocks in the macro block contain coded coefficients, signalling changes in Quantisation Parameter (QP), etc.) followed by coded coefficients for each 8x8 block.
   In the decoder, the sequence of Variable Length Coded (VLCs) is decoded to extract the quantised transform coefficients, which are re-scaled and transformed by an 8x8 IDCT to reconstruct the decoded I-VOP.

2. Predicted Frame: a frame may be predicted (using motion compensation) based on another previously decoded frame. Such frames are called Predicted frame (P-frame);
   A P-frame is coded with Inter prediction from a previously encoded I- or P-frame, called reference frame (Figure. 3.4). The differences between the encoding and decoding stages for an I-frame and for a P-frame are based on the motion estimation and compensation.

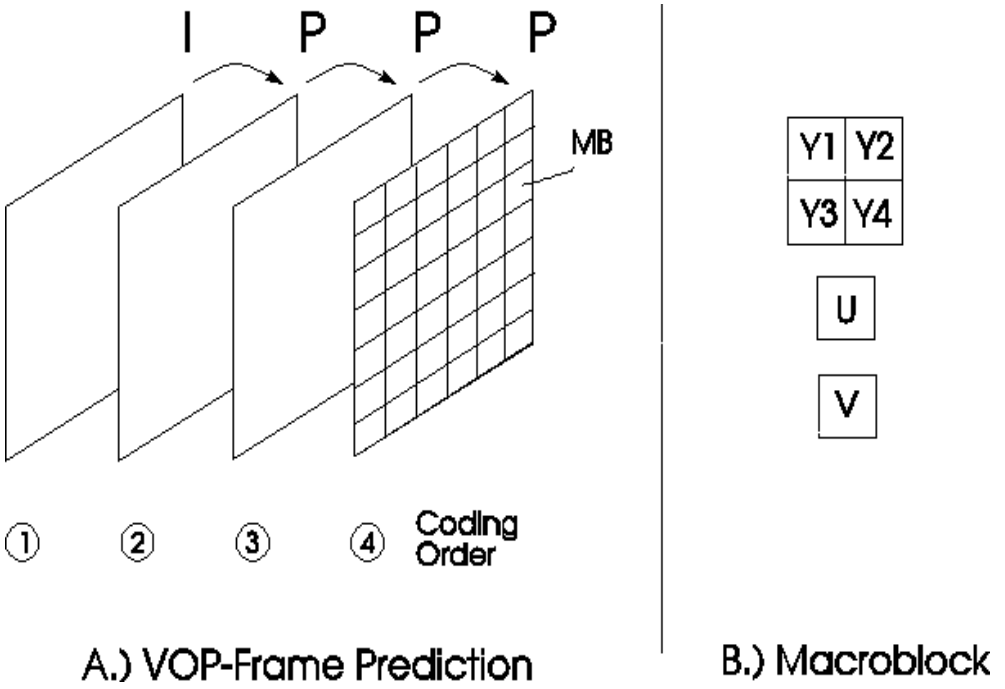

Figure. 3.4/A: Illustration of one I-picture (I-VOP) and three P-picture (P-VOP) coded using inter prediction from the previous frame. Each frame is divided into Macro blocks (MB).

Figure. 3.4/B: With each MB, information related to the colour standard Y:U:V, that is four luminance (Y1, Y2, Y3, Y4) and two chrominance blocks (U, V), is coded. Each of these blocks contains 8x8 pixels.

The basic motion compensation scheme is block-based compensation of 16x16 (luminance) and 8x8 (chrominance) pixel macro blocks. The matching region is subtracted from the current macro block to produce a residual macro block (Motion-Compensated Prediction, MCP).
After motion compensation, the residual data is transformed with the DCT, quantised, reordered, run-level coded and entropy coded. The quantised residual is rescaled and inverse transformed in the encoder in order to reconstruct a local copy of the decoded MB for further motion compensated prediction. A coded P-VOP consists of VOP header, optional video packet headers and coded macro blocks each containing a header (including encoded motion vectors) and coded residual coefficients for every 8x8 block.
The decoder forms the same motion-compensated prediction based on the received motion vector and its own local copy of the reference VOP. The decoded residual data are added to the prediction to reconstruct a decoded macro block (Motion-Compensated Reconstruction, MCR).
Macro blocks within a P-VOP may be coded in Inter mode, with motion compensated prediction from the reference VOP or Intra mode, with no motion compensated prediction. Inter mode will normally give the best coding efficiency but Intra mode may be useful in regions where there is not a good match in a previous VOP, such as a newly-uncovered region.

3. A frame may be predicted based on past as well as future frames as shown in Figure 3.5. Such frames are called Bi-directional Interpolated frame (B-frame). B-frame may only be interpolated based on I-frame or P-frame.

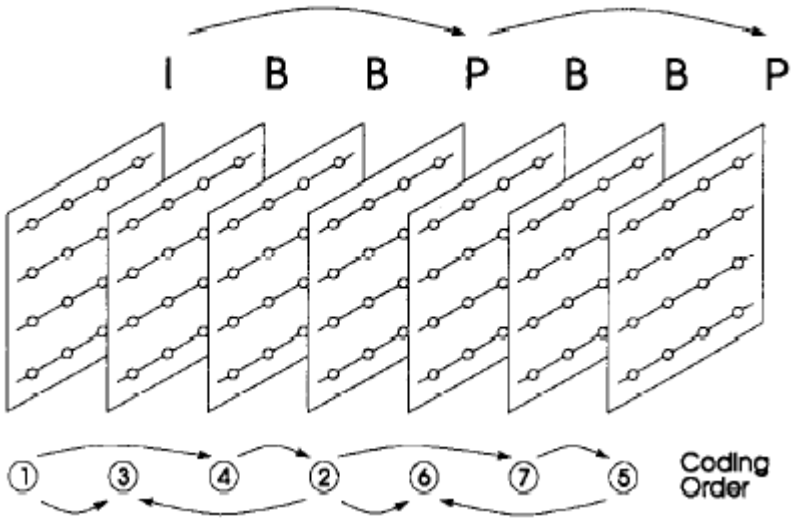

Figure 3.5. Frame prediction

4 MPEG-4 PROFILES AND LEVELS

MPEG-4 provides a large and rich set of tools for the coding of audio-visual objects. In order to allow effective implementations of the standard, subsets of the MPEG-4 Systems, Visual, and Audio tool sets have been identified, that can be used for specific applications. These subsets, called 'Profiles', limit the tool set a decoder has to implement. For each of these Profiles, one or more Levels have been set, restricting the computational complexity. The approach is similar to MPEG-2, where the most well known Profile/Level combination is 'Main Profile @ Main Level'. A Profile@Level combination allows:

- a codec builder to implement only the subset of the standard he needs, while maintaining interworking with other MPEG-4 devices built to the same combination, and
- checking whether MPEG-4 devices comply with the standard ('conformance testing').

Profiles exist for various types of media content (audio, visual, and graphics) and for scene descriptions. MPEG does not prescribe or advise combinations of these Profiles, but care has been taken that good matches exist between the different areas.

4.1 MPEG-4 Part 2: Visual Profiles

The visual part of the standard provides profiles for the coding of natural, synthetic, and synthetic/natural hybrid visual content. There are five profiles for natural video content:

1. **The Simple Visual Profile** provides efficient, error resilient coding of rectangular video objects, suitable for applications on mobile networks, such as Personal Communication Services (PCS) and IMT2000.
2. **The Simple Scalable Visual Profile** adds support for coding of temporal and spatial scalable objects to the Simple Visual Profile. It is useful for applications which provide services at more than one level of quality due to bit-rate or decoder resource limitations, such as Internet use and software decoding.
3. **The Core Visual Profile** adds support for coding of arbitrary-shaped and temporally scalable objects to the Simple Visual Profile. It is useful for applications such as those providing relatively simple content-interactivity (Internet multimedia applications).
4. **The Main Visual Profile** adds support for coding of interlaced, semi-transparent, and sprite objects to the Core Visual Profile. It is useful for interactive and entertainment-quality broadcast and DVD applications.
5. **The N-Bit Visual Profile** adds support for coding video objects having pixel-depths ranging from 4 to 12 bits to the Core Visual Profile. It is suitable for use in surveillance applications.

Version 2 adds the following Profiles for natural video:

6. **The Advanced Real-Time Simple Profile** (ARTS) provides advanced error resilient coding techniques of rectangular video objects using a back channel and improved temporal resolution stability with the low buffering delay. It is suitable for real time coding applications; such as the videophone, tele-conferencing and the remote observation.

7. **The Core Scalable Profile** adds support for coding of temporal and spatial scalable arbitrarily shaped objects to the Core Profile. The main functionality of this profile is object based Signal Noise Ratio (SNR) and spatial/temporal scalability for regions or objects of interest. It is useful for applications such as the Internet, mobile and broadcast.

8. **The Advanced Coding Efficiency** (ACE) Profile improves the coding efficiency for both rectangular and arbitrary shaped objects. It is suitable for applications such as mobile broadcast reception, the acquisition of image sequences (camcorders) and other applications where high coding efficiency is requested and small footprint is not the prime concern.

In subsequent Versions, the following Profiles were added:

9. **The Advanced Simple Profile** looks much like Simple in that it has only rectangular objects, but it has a few extra tools that make it more efficient: B-frames, ¼ pel motion compensation, extra quantization tables and global motion compensation.

10. **The Fine Granularity Scalability Profile** allows truncation of the enhancement layer bit stream at any bit position so that delivery quality can easily adapt to transmission and decoding circumstances. It can be used with Simple or Advanced Simple as a base layer.

11. **The Simple Studio Profile** is a profile with very high quality for usage in Studio editing applications. It only has I frames, but it does support arbitrary shape and in fact multiple alpha channels. Bit rates go up to almost 2 Gigabit per second.

12. **The Core Studio Profile** adds P frames to Simple Studio, making it more efficient but also requiring more complex implementations.

The scenarios in Betsy involve coding of rectangular objects under bandwidth and processing constraints. Therefore the main profiles to consider are the Simple Profile, the Advanced Simple Profile, the Advanced Real-Time Simple Profile, the Advanced Coding Efficiency and the Simple Scalable Profile and the Fine Granularity Scalability Profile if scalability is a requirement.

4.2 MPEG-4 Part 10: Advanced Video Codec (AVC)

The Advanced Video Codec (AVC) is the joint project of the ITU-T/VCEG and ISO/IEC MPEG. This new standardization effort offers both enhanced compression efficiency over existing video coding standards (H.263, MPEG-4 Part 2) and network friendly video representation. Higher compression efficiency is achieved by introducing new coding tools with respect to the previous standards such as multiple reference pictures, seven different block sizes for motion compensation, in-loop deblocking filter, context-based arithmetic coding… The codec is aimed at both conversational (bi-directional and real-time video telephony, videoconferencing) and non-conversational (storage, broadcasting, streaming) applications for a wide range of bit-rates over wireless and wired communication networks. Like previous video coding standards, AVC is based on a hybrid block-based motion compensation and transform-coding model [1] . Additional features improve the compression efficiency and the error robustness at the expense of an increased complexity.

An important concept of AVC is the separation of the system into two layers: a Video Coding Layer (VCL), providing the high-compression representation of data, and a Network Adaptation Layer (NAL), packaging the coded data in an appropriate manner based on the characteristics of the transmission network. The basic coding framework defined by AVC is similar to the one of previous video coding standards: translational block-based motion estimation and compensation, residual coding in a transformed domain and entropy coding of quantized transform coefficients. Additional tools improve the compression efficiency at an increased implementation cost. The motion compensation scheme supports multiple previous reference pictures (up to 5) and a large number of different block sizes (from 16x16 up to 7 modes including 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 pixel blocks). The motion vector field can be specified with a higher spatial accuracy (quarter or eighth-pixel resolution instead of half pixel) and a Rate-Distortion (RD) Lagrangian technique [3] optimizes both motion estimation and coding mode decisions. Since the residual coding is in a transformed domain, a Hadamard transform [1] can be used to improve the performances of conventional error cost functions such as the sum of absolute differences. A deblocking filter within the motion compensation loop aims at improving prediction and reducing visual artefacts. AVC adopts spatial prediction for intra-coding being the pixels predicted from the neighboring samples of already coded blocks. To this aim the standard [1] provides a DC plus 8 directional modes involving linear combinations of the samples. The conventional 8x8 floating-point discrete cosine transform, specified with rounding error margins, is replaced by a purely integer spatial transform basically working on 4x4 shapes. The small sizes help to reduce blocking and ringing artefacts while the integer specification prevents any mismatch between encoder and decoder. Finally, two methods are specified for entropy coding: a Universal Variable Length Coder (UVLC) that uses a single reversible VLC table for all syntax elements and a more sophisticated Context Adaptive Binary Arithmetic Coder (CABAC) [4] .

4.2.1 Profiles and Levels in AVC

One of the key advantages of a standard is that it allows interoperability among the equipments or software developed by different companies and across various applications.

To provide such interoperability, H.264/AVC defines a set of conformance points called Profiles and Levels. All decoders and bit streams compliant with a particular Profile and Level must obey the rules and conditions specified for that Profile and Level.

H.264/AVC contains a rich set of video coding tools. All the coding tools are not required for all the applications. For example, the error resilience tools are not important for the networks with very little errors. Forcing every decoder to implement all the tools will make a decoder unnecessarily too complex. Therefore, subsets of coding tools, with different classes of applications in mind, are defined. These subsets are called Profiles. A decoder may choose to implement only one subset (Profile) of tools. Following three profiles are defined:

. Baseline
. Extended
. Main

Table 4.1 summarizes the coding tools included in these profiles at a high level. A decoder compliant with a particular Profile and Level must implement all the tools specified in that Profile. An encoder may chose to use a smaller subset of tools specified in a Profile to generate a bit stream compliant with that Profile and Level. Baseline profile includes I and P-slices (I and P video packets), Error Resilience tools, and CAVLC. It does not contain B, Spare P (SP) and Spare I (SI)-slices [1] , Interlace coding and CABAC coding tools. It was designed with those applications in mind that run on the platforms with low processing power and in the environment with large packet losses. Among the three Profiles, it has the least coding efficiency. Extended is a super set of Baseline and includes B, SP and SI-slices and Interlace coding tools in addition to all the Baseline Profile's coding tools. It does not include CABAC. It is more complex and provides better coding efficiency than Baseline. Main Profile includes I, P and B-slices, Interlace Coding, CAVLC and CABAC. It does not include Error Resilience Tools and SP & SI-slices. It was designed to provide the highest possible coding efficiency.

Coding Tools	Baseline	Extended	Main
I, P Slices	X	X	X
CAVLC	X	X	X
Error Resilience	X	X	
SP and SI Slices		X	
B Slices		X	X
Interlaced Coding		X	X
CABAC			X

Table 4.1: Profiles in H.264/AVC

4.2.2 Error Resilience Tools

Resilience tools such as resynchronization video packets (slices), intra Macroblock refreshments or data partition are also used in the AVC codec.

The main resilience tools introduced with respect MPEG-4 Part 2 are: (1) Flexible Macro block Order (FMO), (2) Arbitrary Slide Order (ASO), and (3) Redundant Slices. FMO and ASO work to randomize the data prior to transmission, so that if a segment of data is lost (e.g. a packet, or several continuous packets), the errors are distributed more randomly over the video frames, rather than in a single contiguous block of pixels. This helps to preserve more local information in all areas, at the cost of some randomly distributed loss. Redundancy Slices offer more protection by reducing the chance of loss via redundancy, a common approach at the level of channel coding. The additional tools of Data Partitioning and SP/SI slices are also valuable for error resilience/recovery.

4.2.3 MPEG-4 Part 10 Amendment 1: The Scalable Video Codec (SVC)

The scalable video coding (SVC), also called Joint Scalable Video Model (JSVM), is currently being developed as an extension of the ITU-T Recommendation H.264 | ISO/IEC International Standard ISO/IEC 14496-10 advanced video codec (AVC). The intent is to create a standard for efficient video compression that provides bit streams scalable in frame rate, resolution and SNR quality. The SVC is an extension to the Advanced Video Coding that will add enhanced forms of quality scalability to further enable advanced video coding uses in a wide variety of applications, particularly including highly-heterogeneous environments.

There are two different ways of introducing scalability in a codec, either by using a technique that is intrinsically scalable (such as bit plane arithmetic coding) or by using a layered approach (same concept as the one that is used in many previous standards [9]). Here, a combination of the two approaches to enable a full spatio-temporal and quality scalable codec is used. Temporal scalability is enabled by Motion Compensated Temporal Filtering (MCTF) [6] , whereas spatial scalability is provided using a layered approach. For quality (SNR) scalability, two different possibilities are provided; an embedded quantization approach for coarse grain scalability and a fine grain scalability (FGS) approach based on the principle of sub-bit plane arithmetic coding.

The base layer is encoded using a single layer AVC coding scheme. The motion estimation and mode decision process is performed using the Lagrangian optimisation techniques as described in the case the JM of AVC [5], with a few additions to handle the scalability aspects. The encoding of the motion and the residual texture is based on AVC [5], with few modifications to handle the spatial and SNR scalability aspects.

For each spatial layer a motion compensated temporal decomposition is performed. This decomposition provides temporal scalability. Motion information from lower spatial layers can be used for prediction of motion on the higher layers. For texture encoding, spatial prediction between successive spatial layers can be applied to remove redundancy.

The residual signal resulting from intra prediction or motion compensated inter prediction is transform coded. A quality base layer residual provides minimum reconstruction quality at each spatial layer. This quality base layer can be encoded into an AVC compliant stream if no inter layer prediction is applied. For quality scalability, quality enhancement layers are additionally encoded. These enhancement layers can be chosen to either provide coarse or fine grain quality (SNR) scalability.

4.3 Codec Selection

The different profiles present different trade-offs between coding efficiency (bit rate), complexity and quality. This way, while some profiles achieve higher compression efficiency at the cost of an increased complexity other profiles aim at a lower complexity in processing constrained environments and trade off some coding efficiency.

The **MPEG-4 Simple Profile** is the "basis" of all profiles and presents a good tradeoff in terms of implementation complexity, while still providing an acceptable coding efficiency and error resilient features. Thus, it is mainly designed for low processing power coding and low latency, being suitable for real-time encoding in mobile and wireless devices, video telephony and video surveillance, applications, which coincide with Betsy's scenarios (video chat in home network and hot spot scenario with cameras).

As for its implementation complexity, Imec has an in-depth expertise in implementation issues related to MPEG-4 Simple Profile [12] [13] can provide accurate and realistic energy measurements.

To further increase the coding efficiency of the Simple Profile, new coding tools are introduced in the **Advanced Simple Profile** such as B frames, Global Motion Compensation and Quarter-pel Motion Compensation and further tools in the **Advanced Coding Efficiency Profile**. These profiles target more demanding environments in terms of processing capabilities and bit rate and they come with a complexity and power increase at both encoder and decoder.

The Advanced Video Codec (AVC or MPEG-4 Part 10) enhances the compression efficiency and error resilience with respect to other video coding standards such as H.263, MPEG-4 Part 2 and provides network friendly video representation. This is done also at the expense of an increased complexity. A comparison between the AVC and the MPEG-4 Simple Profile coding efficiency and complexity can be found in [24] ,[25] .The analysis done in these documents shows that even in its most simple configuration using similar tools as in the MPEG-4 SP except for one-fourth pixel motion estimation and in-loop deblocking filter) the encoder complexity roughly doubles with respect to the MPEG-4 Simple Profile and the memory requirements are also highly increased. This way, for the Betsy scenarios and power-constrained devices the compression efficiency provided at low complexity by MPEG-4 Part 2 codec is a reasonable choice.

When transmission errors occur, the video quality degrades rapidly and the use of error resilience tools becomes a key factor to overcome quality degradation. The Simple Profile provides some low complexity error resilience tools. To further increase the error resilience the **Advanced Real Time Simple Profile** provides two new resilience tools, namely the dynamic Resolution Conversion and the NEWPRED (for 'new prediction') technique, while keeping the same coding tools as the Simple Profile. These resilience tools need a timely feedback from the decoder to the encoder in order to provide with efficient adaptation.

In our scenarios the availability of timely feedback can pose some challenges.

The concept of timely feedback is relative, and depends on what info is fed back, what resilience techniques are used, the adaptation speed...etc. This way, a faster feedback might or might not be needed.

For the NEWPRED technique you need to feedback that a particular frame was received erroneously so that the encoder predicts from another one, obviously the feedback there should be as fast as possible to stop error propagation (1 frame).

The coherence time of the channel can be around 100 ms (3 or 4 frames) so feedback should be faster if adaptation to channel conditions is performed. If the channel condition is feedback for example, we would like the feedback to happen fast, otherwise in 3 frames the channel condition to which we adapt (for instance by changing the QP, or the resilience strength), may have already changed.

In the Home Network scenarios 1 and 2 of Betsy, as the content is pre-encoded, no feedback information can be exploited unless some transcoding takes place previously in the media center. Transcoding can be envisaged since energy awareness is primarily oriented to the mobile devices and thus the computation/energy cost of e.g transcoding can be moved to the media center or base station.

The Home Network scenario 3 consists of a video conversation where the feedback information can be piggybacked in the opposite flow. There are two hops (being each hop the link between device and AP), where the feedback delay depends mainly on the delay to access the medium in each hop, while the propagation delay is negligible. The delay at each hop can be lower than 10 ms (dependent on the network load), which makes a total delay of around 40 ms since the information is sent (through 2 hops) until the feedback is received (other 2 hops back), which causes a reaction within at least 2 video frames delay. In the hot spot scenario there is need for a feedback channel from the decoder to the camera encoders, where the feedback delay of the transmission through the wired backbone (connection between hot spot AP and network AP) needs to be added to the previous one of around 40 ms.

Another issue is that the introduction of these techniques increases the processing and memory requirements at both encoder and decoder sides. The NEWPRED technique requires the buffering of multiple frames in memory so as to allow the prediction from different frames, which increases the memory footprint at both encoder and decoder. Similarly, in the Dynamic Resolution Conversion technique both down sampling and up sampling are needed plus it is necessary to estimate the motion vector for both resolutions in an efficient way.

The AVC codec provides as well new error resilience tools as Flexible Macro block Ordering (FMO), Arbitrary Slice Order (ASO) or Redundant Slices. These increased error resilience comes again at a complexity increase. Nevertheless, if a need for higher compression efficiency or more powerful resilience tools was needed and the extra complexity could be justified the use of AVC in a simple configuration could be explored. The Simple configuration [25] sets the encoding tools to a similar behavior as the MPEG-4 simple profile. Only the fourth pixel motion vectors and the in-loop deblocking cannot be disabled.

The single layer codecs seen so far can adapt to the bandwidth and complexity constraints by modifying its coding and resilience parameters. Chapter 3 describes the impact of the adaptation of the coding parameters on the quality, bit rate and complexity axes. This provides some sort of scalability that may be sufficient to cope with varying bandwidth and complexity constraints in most Betsy scenarios.

However, true scalability refers to the use of multiple layer codecs that provide a bit stream with multiple layers of information associated with different quality, complexity and bit rate levels. The amount of layers processed by the decoder determines the end quality and associated rate and complexity.
Both the **Simple Scalable Profile** and the **Fine Grain Scalability Profile** can provide this multi-layered scalability.

The Simple Scalable Profile provides with respect to the Simple Profile the use of B frames and both temporal and spatial scalability. The spatial scalability is achieved by switching to a lower spatial resolution, this way, the Base layer provides with QCIF resolution while the enhancement layer provides a higher CIF resolution. Temporal scalability is performed by dropping some frames (Enhancement layers) reducing then the temporal resolution (frames per second) without impairing the decoding of the base layer with reduced resolution.

The current Sw and Hw optimized implementation of the MPEG-4 SP of Imec can easily be extended with temporal and spatial scalability without a big impact on the codec complexity.
The **Fine Grain Scalability Profile** can further extend the possibilities of scalability by providing a scalable bit stream where the decoding process can stop at any time. This finer degree of scalability comes at the cost of a lower coding efficiency than a single layer codec (around 30%) and increased complexity**.** When increased flexibility is required, fine-grained scalability (FGS) can be built on top of a coarse grained system. Moreover, FGS performances largely depend on the quality of the reference base layer [14] **.** Until sufficient quality has been obtained for the base layer, FGS provides worse coding efficiency than the non-scalable simple hybrid MC/DCT scheme.

The Scalable Video Codec is the new MPEG standard that will provide spatial, temporal and fine grain quality scalability. This standard, to be finalized around June 2006, is based on the existing AVC codec, with increased complexity with respect to AVC.

However, in our scenarios true scalability cannot be motivated by the need to adapt to multiple heterogeneous devices receiving the same content simultaneously. The need for scalability comes therefore from adaptation to varying bandwidth and processing capabilities for a single device. In this case, if the content is received by a single device then a single layer encoder can adapt and scale its output to the available bandwidth or processing needs of the decoding device.

In the first and second Home Network scenarios, content stored in the PSC is transmitted to the home devices, though not to different devices simultaneously. If the content is pre-encoded and does not suffer any further transcoding then the adaptability to limited decoding capabilities or channel capacity (change from AP1 to AP2 or when Betsy goes to the garden increased distance from AP…) needs to be provided by the pre-encoded scalable content or storage of independent versions with several resolutions. In the case of pre-encoded Simple Scalable Profile the scalability provided is very coarse, adapting only to large variations of bandwidth and power. Another possibility is to perform in the media center transcoding of the pre-encoded content stored in the PSC (decoding plus encoding adapting with more granularity to the bandwidth and processing constraints).

The third Home Network scenario is a video conversation between Betsy and her mother. Adaptability to varying channel capacity and processing capabilities is needed. This can be performed by means of single layer adaptation or with scalability extensions (temporal and spatial) extending the MPEG-4 SP to a Simple Scalable Profile. Nevertheless, it has to be noted that the scalability provided with the multiple layer Simple Scalable codec is a coarse scalability that adapts to bigger variations of bandwidth and processing requirements. Additionally, the use of multiple layers in the Scalable Profile involves a loss in coding efficiency with respect to the Simple Scalable Profile. A finer adaptation to varying conditions can be achieved by adaptation of the coding parameters of a single-layer codec. Alternatively, fine adaptation can be obtained with Fine Grain Scalability but at the cost of an increased complexity and reduced efficiency.

In the Hot Spot scenario of Betsy each camera encodes the content and sends it to the device in the home network (in our case a tablet pc or a big TV at home). The processing limitations will most probably come at the encoder side (wireless camera) together with limited bandwidth for the uplink communication with the hot spot AP, which is being shared by multiple cameras. In this case, a single layer codec (MPEG-4 SP) would be able to scale down its output to the limited bandwidth and encoder processing resources while being less power demanding than other profiles (easier to encode at real time keeping low power consumption).

For this reason, the Simple Profile, even without scalability extensions (Simple Scalable Profile), can provide with enough adaptability in our scenarios while maintaining a good tradeoff in terms of coding efficiency and resilience capacities at a low implementation complexity. Therefore, this is the preferred profile if no additional features are really required. However, if scenarios state a clear requirement to extended features from other profiles, the complexity implications should first be rigorously evaluated, as energy-awareness is an important issue in BETSY.

5 Functional attributes of MPEG-4

This section presents the impact of the main coding parameters of the MPEG-4 Simple Profile codec on the quality, rate and complexity axes. These coding parameters are the quantization parameter (QP), which determines the encoding accuracy by setting a finer or coarser quantization step, the temporal resolution (in terms of video frames per second) and the spatial resolution. The value of these parameters has a direct impact on the quality, bit rate and codec complexity axes determining a particular tradeoff in terms of quality, bit rate and complexity at encoder/decoder. Therefore, the tuning of these parameters allows adapting to the available bandwidth and processing resources.

5.1 Performance in terms of Rate – Distortion

The coding efficiency of a video codec is given by its Rate – Distortion graph in which the quality and compression tradeoffs can be seen.

Figure 5.1 to Figure 5.4 present the Rate – Distortion tradeoffs for three well- known video sequences: *Mother and Daughter* (low complexity, head and shoulders sequence), and *Calendar and Mobile* (high complexity, with a lot of movement).

All sequences are compressed using a proprietary MPEG-4 video encoder[12] **.** The rate control has been disabled to exclude the QP variation along the sequence.

Quality, measured in PSNR (Peak Signal to Noise Ratio) and the associated bit rate are given for different spatial resolutions (CIF, QCIF), temporal resolution (number of fps, frames per second) and quantization parameters (QP). Table 5.1 gives a possible conversion between PSNR and MOS (Mean Opinion Score) that gives the human quality impression on a scale from 5 (best) to 1 (worst).

PSNR [dB]	MOS
> 37	5 (Excellent)
31 - 37	4 (Good)
25 - 31	3 (Fair)
20 - 25	2 (Poor)
< 20	1 (Bad)

Table 5.1: Possible PSNR to MOS conversion [26]

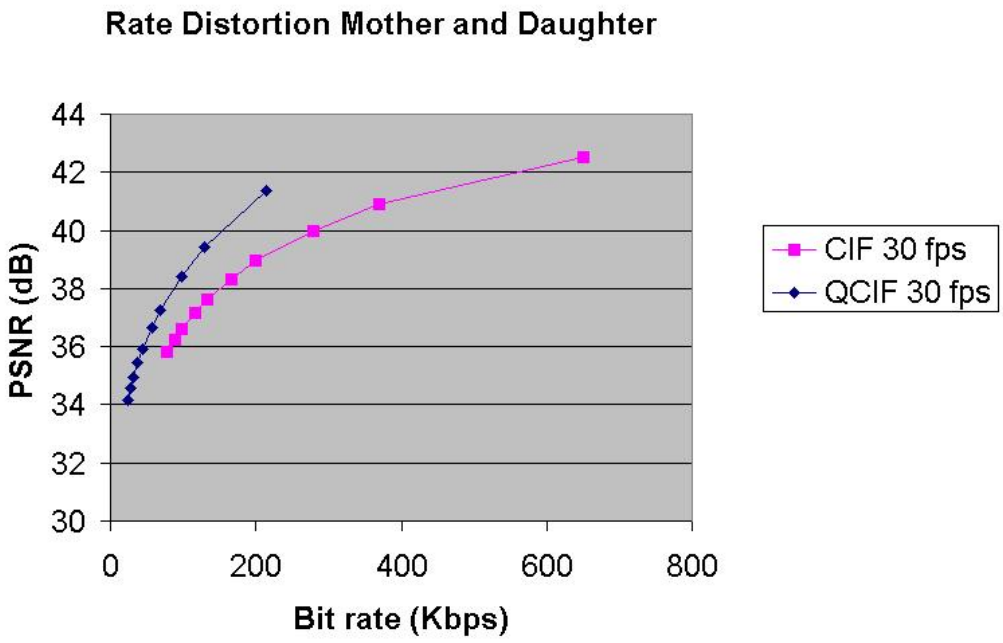

Figure 5.1. Rate Distortion performance for Mother and Daughter at 30 fps.

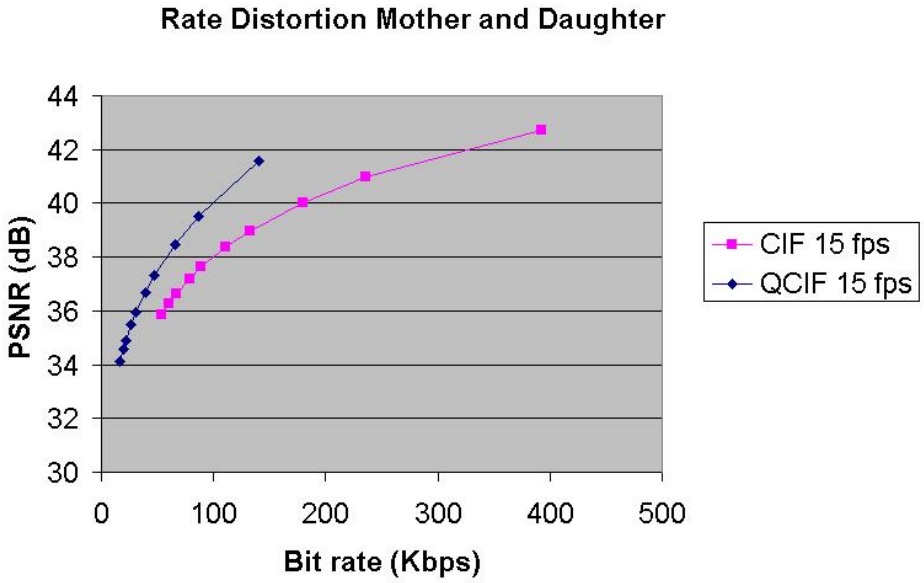

Figure 5.2. Rate Distortion performance for Mother and Daughter at 15 fps.

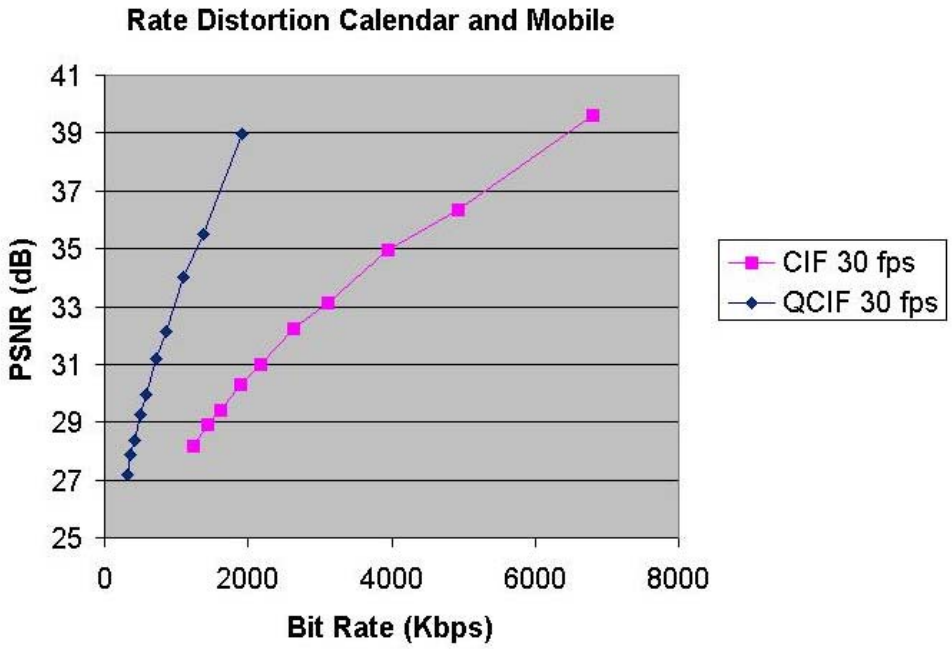

Figure 5.3. Rate Distortion performance for Calendar and Mobile at 30 fps.

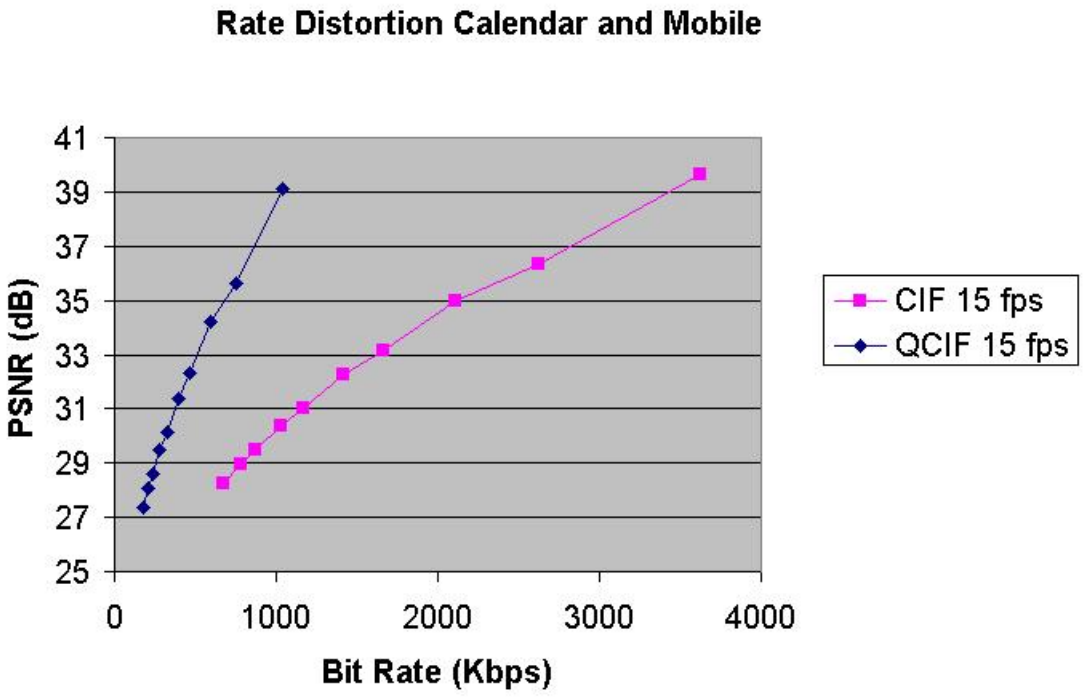

Figure 5.4 Rate Distortion performance for Calendar and Mobile at 15 fps

It needs to be noted that current quality metrics, such as the widely used PSNR, can hardly reflect the effect of a reduced temporal resolution or in other words the presence of motion jerkiness. That is the reason why full temporal resolution sequences (30 fps) and those of reduced temporal resolution (15 fps) present similar objective quality. Generally, the lower the temporal resolution the more noticeable the motion jerkiness becomes. Standard televisions display frames at a rate of 30 frames per second, while current High Definition Televisions go up to 60 frames per second. The perception of motion jerkiness for a particular frame rate depends on the video content. Sequences with little or slow motion can reduce its frame rate more than sequences with high and sudden motion peaks where the jerkiness will be more noticeable. However, as a rule of thumb, for frame rates above 10 fps the motion jerkiness is acceptable.

Another well-known problem of objective quality metrics is that they do not always correlate well with subjective visual quality. In particular when transmission errors occur and the PSNR values decrease the correlation between the PSNR and the subjective quality decreases. Quality metrics like the Structural Similarity Index (SSIM) [15] ,[16] improve the correlation with subjective quality but still fail to reflect accurately the effect of transmission errors.

5.2 Bandwidth and Complexity Scaling

The following paragraphs show the impact of the coding parameters on the output bit rate.

5.2.1 Bandwidth Scaling

5.2.1.1 SNR Scaling: QP

The Quantisation Parameter (QP) controls the accuracy of the texture information. A high QP lowers the number of bits required to code a texture block at the cost of precision. Figure 5.5 a,b display the influence of the QP at different frame rates and spatial resolutions (QCIF and CIF). The effect of a higher QP on the image quality is that the image becomes blurrier, due to the coarser quantization. In general, QP 3 to QP 6 can be very good quality, QP 7 to 10 is still fair quality, and QP 11 and 12 start to be blurrier. Nevertheless, the impact of QP on the visual quality is still content dependent.

On Figure 5.5a,b the absolute bit rate depends on the sequence, but its relative decrease is quasi independent of both the sequence type (complex or not) and resolution (QCIF or CIF). A major trend to notice on these graphs is the exponential decrease of the bit-rate with QP. This drop is particularly fast and significant at high frame rates. At lower frame rates, the higher relative amount of information contained in the motion vectors reduces the impact of QP on the bit stream size.

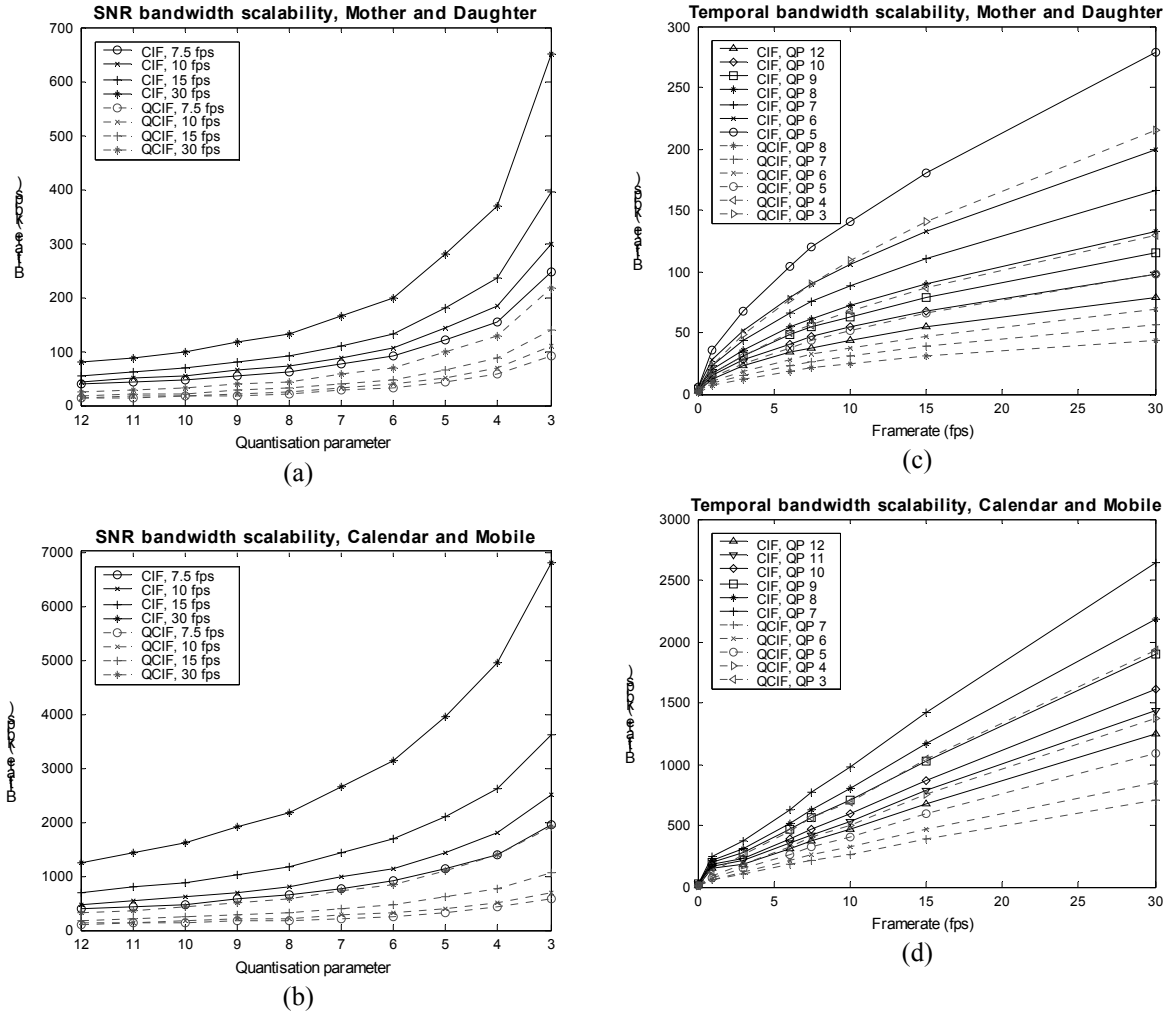

Figure 5.5: Bandwidth scaling by QP increasing (a,b) and by lowering the frame rate (c,d) has respectively an exponential and quasi-linear impact. The results are plotted for the Mother and Daughter (a,c) and Calendar and Mobile (b,d) video sequences.

5.2.1.2 Temporal Resolution Scaling: Frame rate

Lowering the frame rate decreases the bit rate. Due to accuracy/efficiency loss of the Motion Estimation (ME) at low frame rates and due to the variable length, predictive motion vector encoding method, the reduction is expected to be non-linear. Figure 5.5.c,d illustrate the bandwidth scaling with temporal resolution for a set of sequences, encoded with different QP and spatial resolution (QCIF or CIF):

- A high quantisation level masks the temporal scaling effect (smaller slope of the curve). In that case, the motion description part dominates the bit stream.
- For complex sequences, increasing the frame rate has a higher relative cost than for simple and temporally correlated sequences. However, this increase is never more than linear.
- The graph is close to a linear at high frame rates.
- The spatial resolution (QCIF and CIF) has no influence on the relative bit rate reduction.

5.2.1.3 Spatial Resolution Scaling: QCIF or CIF

The effect of the frame size reduction (from CIF to QCIF) on the bit rate is illustrated in Table 5.2. The bandwidth drop slightly increases with the frame rate and the complexity of the sequence. Actually, QCIF frames are obtained by sub-sampling spatially filtered CIF sequences (to avoid aliasing). Consequently, the relative amount of information per encoded macro block is increased at lower resolution and hence, the bit rate drop is below the expected factor four. For low complexity sequences containing few high frequency spatial components, the filtering removes few information and the relative bit rate drop is lower. A higher quantisation parameter increases the effect of the frame size reduction on the bit rate as it removes more of this relative increment of high frequency components.

Table 5.2: Spatial total bandwidth reduction factors Mother and Daughter and Calendar and Mobile

fps \ QP	3.00	6.00	7.50	10.00	15.00	30.00
12	3.2	3.2	3.3	3.2	3.2	3.3
11	3.1	3.1	3.2	3.2	3.1	3.2
10	3.1	3.1	3.1	3.1	3.1	3.1
9	3.0	3.0	3.0	3.0	3.0	3.1
8	2.9	3.0	2.9	2.9	2.9	3.0
7	2.9	2.9	2.8	2.8	2.8	2.9
6	2.9	2.8	2.8	2.8	2.8	2.9
5	2.8	2.7	2.7	2.7	2.7	2.9
4	2.7	2.7	2.7	2.7	2.7	2.9
3	2.8	2.7	2.7	2.8	2.8	3.0

fps \ QP	3.00	6.00	7.50	10.00	15.00	30.00
12	3.7	3.6	3.6	3.8	3.8	3.9
11	3.6	3.6	3.6	3.7	3.7	3.9
10	3.6	3.6	3.6	3.7	3.7	3.9
9	3.6	3.6	3.5	3.7	3.7	3.8
8	3.6	3.5	3.5	3.6	3.6	3.8
7	3.5	3.5	3.5	3.6	3.6	3.7
6	3.5	3.5	3.5	3.6	3.6	3.7
5	3.5	3.5	3.5	3.6	3.5	3.6
4	3.5	3.5	3.4	3.5	3.5	3.6
3	3.5	3.5	3.5	3.6	3.5	3.5

5.2.2 Complexity Scaling

The tuning of the video coding parameters not only determines an operating point in the Rate Distortion performance but it also determines the associated encoding and decoding complexity. This way, by tuning the coding parameters the encoding and decoding complexity can be adapted to processing constraints. In this section we present how the complexity of a hybrid Motion Compensated/Discrete Cosine Transform (MC/DCT) video encoder/decoder (H.16x/MPEG) scales with the encoding parameters.

The evaluation of the complexity of the hybrid MC/DCT video coded is based on time measurements of a proprietary MPEG-4 simple profile codec [12] ,[18] optimized towards memory [19] **.** As video coding/decoding are data dominated systems, efficient memory management and data transfer and storage optimization directly impact their resource requirements. The performances are measured as coding speeds on a Pentium III running at 700 MHz with Windows NT and are expressed in relative times, with the duration of the sequence in seconds as reference. Meeting real-time constraints means having a relative coding time smaller than one.

The close relation between the memory load and the speed performance has been evaluated for test cases with different coding parameters in [12] **,** justify assessing the complexity of the system by measuring the relative coding/decoding time fluctuations on a specific platform. The reported resource scaling is expected to be representative for any platform.

Figure 5.7 .a to .d describe the complexity scaling of the MPEG-4 codec as a function of the quantisation parameter.

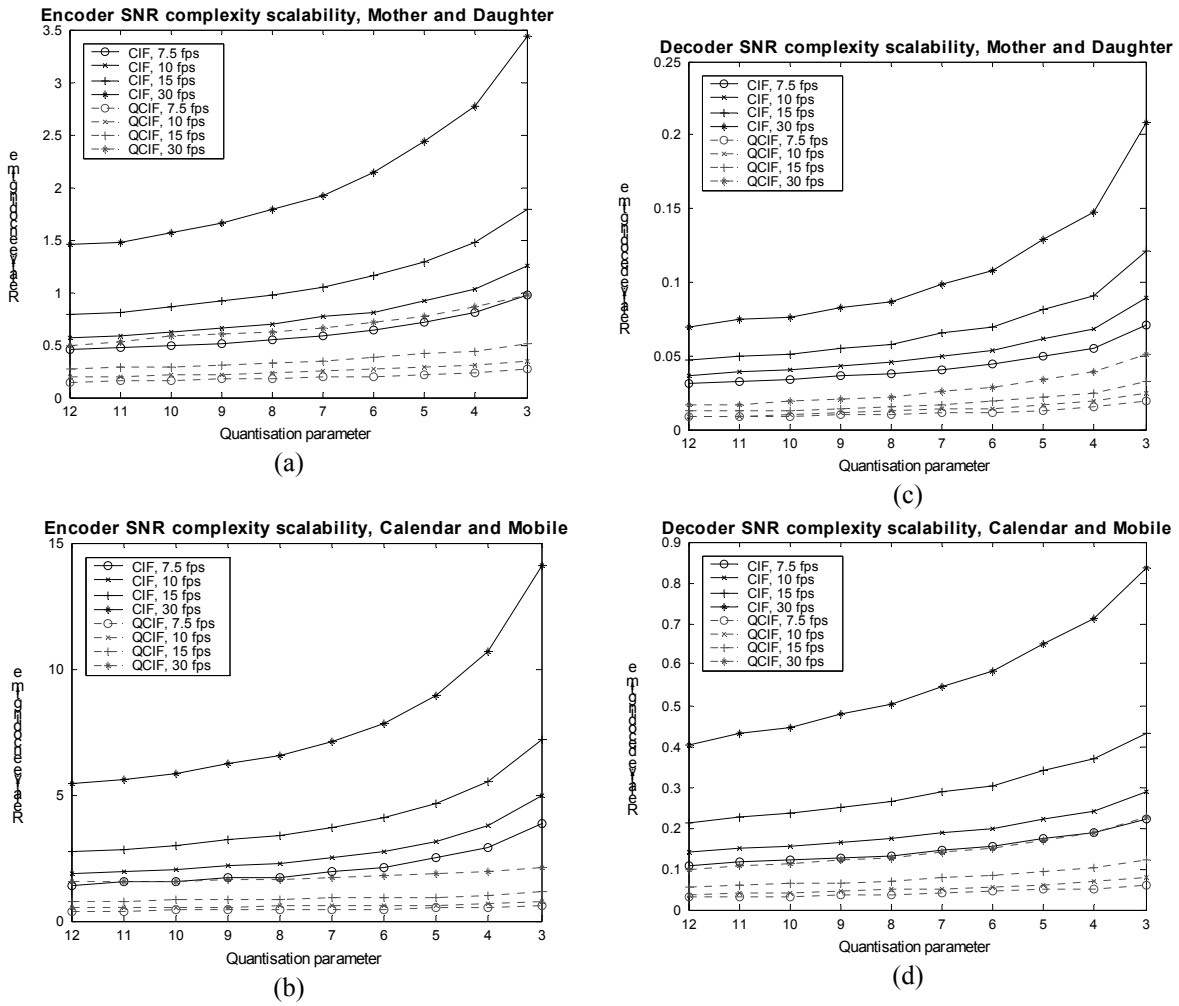

Figure 5.6: Complexity scaling by QP increasing for the encoder (a,b) and the decoder (c,d) has an exponential impact. The results are plotted for the Mother and Daughter (a,c) and Mobile (b,d) video sequences.

5.2.2.1 SNR Complexity Scaling

As for the bandwidth, the complexity decrease is exponential with QP and increases with the frame rate. The nature of the sequence does not really impact it. Only at very small QP (high quality), a faster decrease of the decoding complexity for simple and stationary video occurs. For those sequences, the increase of QP results in lots of zero texture blocks, free of IDCT computation.

The spatial resolution does not affect the relative decrease of the decoder complexity. However, on the encoder side, the impact of QP on the total QCIF encoder complexity is limited.

5.2.2.2 Temporal complexity scaling

The encoder complexity has an almost linear increase with the frame rate. Consequently, encoding two independent streams at half the reference frame rate requires about the same resources as encoding a single stream at the reference frame rate.

For simpler video sequences, the complexity scaling of the decoder is not linear. The decoder optimizations restrict the number of the texture reconstructions (inverse quantisations and IDCTs) to the non-zero texture blocks. This tempers the decoder complexity increase at higher frame rates (smaller slope) for simple sequences.

In contrast to the bandwidth case, the QCIF complexity of both encoder and decoder is far below the one for CIF. The distance between QCIF and CIF complexity is larger for sequences with a high degree of motion. The temporal scaling is independent of the resolution and the quantisation degree.

5.2.2.3 Spatial Complexity Scaling

Switching from CIF to QCIF resolution requires approximately four times fewer resources to decode video, independent of the frame rate and the complexity of the sequence. The factor is directly proportional to the number of processed blocks.

5.2.3 Single Layer Adaptation versus Layered scalability

As discussed previously adaptation to varying bandwidth and processing constraints can be performed by means of single layer adaptation as well as by using a true scalable bit stream consisting of multiple layers. Single layer adaptation involves tuning of the coding and resilience parameters of the bit stream achieving a fine adaptation of the quality, rate and complexity requirements. Adaptation by means of the Quantisation Parameter is named SNR scalability, while spatial scalability involves changes in the spatial resolution and temporal scalability implies variation of the frame rate. This adaptation needs to be performed on a one-to-one communication as simultaneous adaptation to multiple receivers with different requirements cannot be performed. This can only be achieved by sending separate customized bit streams to each receiver (simulcast) or by using a common layered bit stream and letting each receiver decode as many layers as its bandwidth and processing requirements allow.

On one hand, the advantages of a single-layered bit stream are its higher compression efficiency and reduced complexity. On the other hand, a scalable-layered bit stream allows adaptation to different bandwidth and processing constraints simultaneously on different devices. This does not require timely tuning of the encoding parameters providing adaptation with multi-layered pre-encoded content.

However, in some cases, providing a simulcast of several single layer bit streams with different quality levels can provide several scaled solutions of the video content in a more efficient way than with a scalable bit stream. Some studies show that the use of simulcast can be more efficient in terms of bandwidth and complexity than layering approaches for coarse scalability [21] ,[17] .

Depending on the application addressed and on the relative importance of quality, bit rate or complexity factors one of the solutions can be more convenient.

5.3 Performance Comparison between MPEG-4 Simple Profile and Simple Scalable Profile

Figure 5.7 , Figure 5.8. and Figure 5.9 compare the coding efficiency of the MPEG-4 Simple Profile and the Simple Scalable Profile. The scalable codec using temporal scalability has practically the same coding efficiency as a single layer codec as the only loss of prediction efficiency is due to predicting from frames further in time. However, Figure 4.7 shows that the use of spatial scalability clearly involves a loss of coding efficiency, over 30% with respect to the higher resolution (CIF) in the single layer solution. In fact, the bit rate of the scalable bit stream is comparable to that one of encoding two separate bit streams of QCIF and CIF resolutions for simulcast.

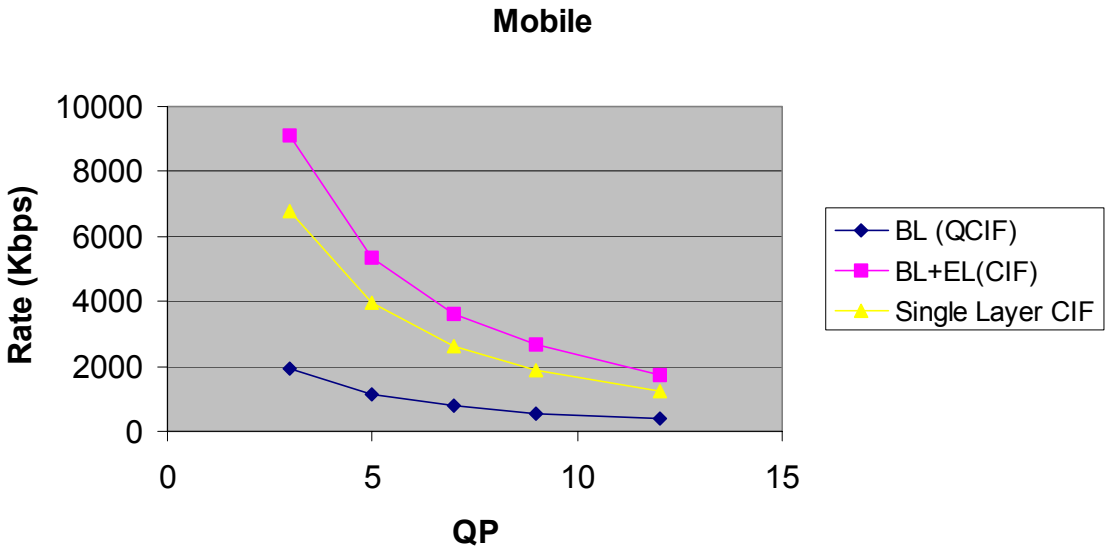

Figure 5.7 Coding efficiency loss in spatial scalability.

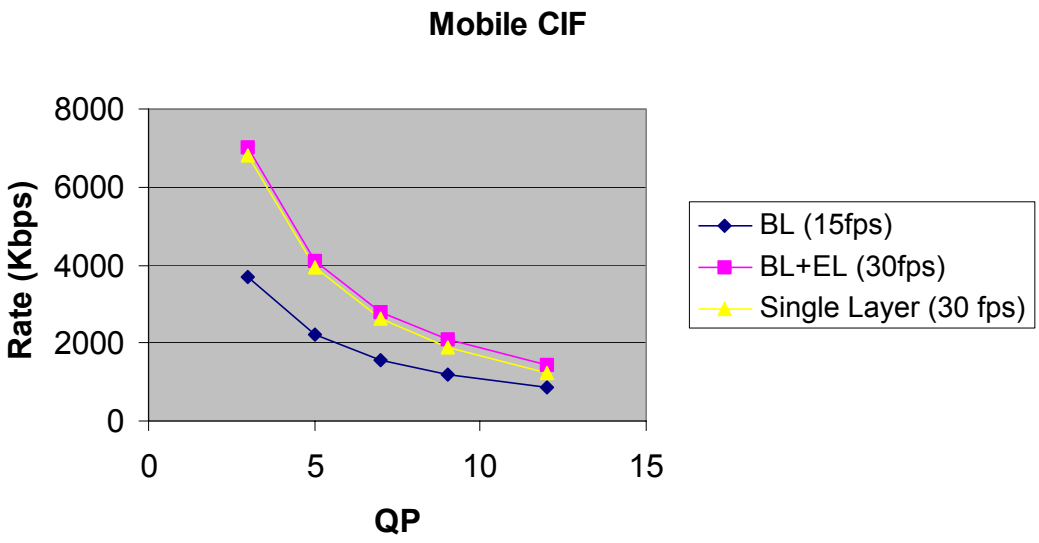

Figure 5.8. Temporal scalability versus single layer for CIF resolution.

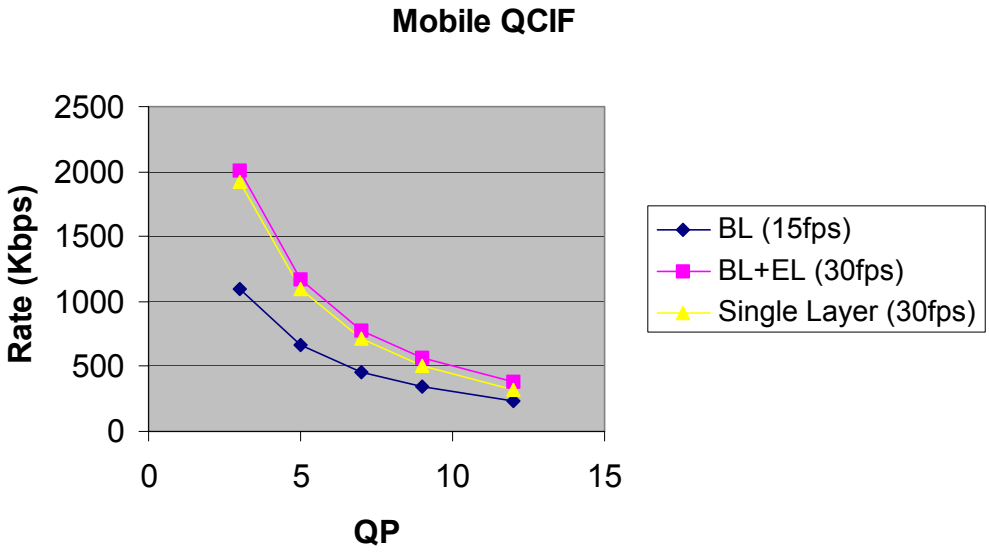

Figure 5.9 Temporal scalability versus single layer for QCIF resolution.

5.4 Error Resilience Tools

MPEG-4 provides error robustness and resilience to allow accessing image or video information over a wide range of storage and transmission media. In particular, due to the rapid growth of mobile communications, it is extremely important that access is available to audio and video information via wireless networks. This implies a need for useful operation of audio and video compression algorithms in error-prone environments.

5.4.1 Resynchronization Video Packet

Resynchronization tools attempt to enable resynchronization between the decoder and the bit stream after a residual error or errors have been detected. Generally, the data between the synchronization point prior to the error and the first point where synchronization is reestablished is discarded. If the resynchronization approach is effective at localizing the amount of data discarded by the decoder, then the ability of other types of tools that recover data and/or conceal the effects of errors is greatly enhanced.

The video packet approach adopted by MPEG-4 is based on providing periodic resynchronization markers throughout the bit stream. In other words, the length of the video packets are not based on the number of macro blocks, but instead on the number of bits contained in that packet. If the number of bits contained in the current video packet exceeds a predetermined threshold, then a new video packet is created at the start of the next macro block.

A resynchronization marker is used to distinguish the start of a new video packet. This marker is distinguishable from all possible VLC codewords as well as the VOP start code. Header information is also provided at the start of a video packet. Contained in this header is the information necessary to restart the decoding process and includes: the macro block number of the first macro block contained in this packet and the quantization parameter necessary to decode that first macro block. The macro block number provides the necessary spatial resynchronization while the quantization parameter allows the differential decoding process to be resynchronized.

Also included in the video packet header is the header extension code. The HEC is a single bit that, when enabled, indicates the presence of additional resynchronization information; including modular time base, VOP temporal increment, VOP prediction type, and VOP F code. This additional information is made available in case the VOP header has been corrupted.

It should be noted that when utilizing the error resilience tools within MPEG-4, some of the compression efficiency tools are modified. For example, all predictively encoded information must be confined within a video packet so as to prevent the propagation of errors.

In conjunction with the video packet approach to resynchronization, a second method called fixed interval synchronization has also been adopted by MPEG-4. This method requires that VOP start codes and resynchronization markers (i.e., the start of a video packet) appear only at legal fixed interval locations in the bit stream. This helps avoiding the problems associated with start codes emulations. That is, when errors are present in a bit stream it is possible for these errors to emulate a VOP start code.

In this case, when fixed interval synchronization is utilized the decoder is only required to search for a VOP start code at the beginning of each fixed interval. The fixed interval synchronization method extends this approach to be any predetermined interval.

The use of independently decodable video packets, also called slices, reduces the effect of network errors at the cost of a reduction of the coding efficiency [22] . The more resynchronization packets are used the higher is the probability of resynchronization within a frame when errors occur. At the same time, the probability of errors occurring in a packet diminishes as the packet becomes smaller. On the other hand, the coding efficiency is reduced as the slices break the intra prediction in a frame and the introduction of resynchronization markers between slices slightly increases the bit rate. The impact on the encoding/decoding complexity is negligible.

5.4.2 Reversible Variable Length Coding (RVLC)

After synchronization has been reestablished, data recovery tools attempt to recover data that in general would be lost. These tools are not simply error correcting codes, but instead techniques that encode the data in an error resilient manner. For instance, one particular tool is Reversible Variable Length Codes (RVLC). In this approach, the variable length codewords are designed such that they can be read both in the forward as well as the reverse direction.

Generally, if a burst of errors has corrupted a portion of the data, all data between the two synchronization points would be lost. However, as shown in Figure 5.10 an RVLC enables some of that data to be recovered. It should be noted that the parameters, QP and HEC shown in the Figure, represent the fields reserved in the video packet header for the quantization parameter and the header extension code, respectively.

The use of RVLC is expected to have a bigger impact on both the coding efficiency and the implementation complexity.

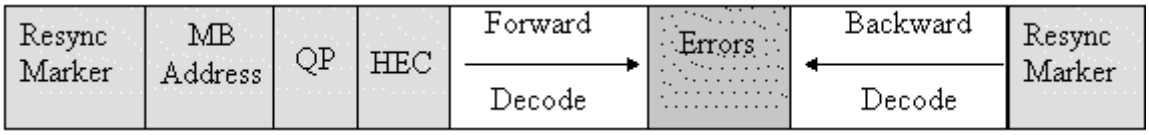

Figure 5.10 Example of Reversible Variable Length Code

5.4.3 Forced Intra Refresh (CIR and AIR)

The use of inter-coding causes temporal error propagation as the current frame may be predicted from previous corrupted frames. This propagation can be suppressed by inserting intra-coding macro blocks, which do not utilize temporal correlation. In many video coding standards, the least use of intra-coding is approved, because the decoded video is gradually degraded by the accumulation of inverse transform mismatch error with the continuous use of inter-coding. This intra refresh serves as a means for error concealment.

Forced intra refresh is a non-normative tool, which consists of a number of macro blocks in each frame being transmitted in intra mode. Two techniques can be used: Cyclic Intra Refresh (CIR), which restores cyclically all the macro block positions within the frame, and Adaptive Intra Refresh (AIR), which attempts to optimise performance by providing a higher rate of intra update in areas of high motion.

This increased error resilience comes also at the cost of an increased bit rate [22] ,[23] . This involves that for a constant bit rate, with the introduction of Intra Macro blocks the video quality is decreased. In terms of complexity the encoding and decoding of Intra Macro blocks is less demanding as motion estimation is not involved, therefore the introduction of Intra Macro blocks slightly decreases the codec complexity.

5.4.4 Error Concealment

Error concealment is an extremely important component of any error robust video codec. Similar to the error resilience tools discussed above, the effectiveness of an error concealment strategy is highly dependent on the performance of the resynchronization scheme. Basically, if the resynchronization method can effectively localize the error then the error concealment problem becomes much more tractable. For low bit rate applications, low delay applications the current resynchronization scheme provides very acceptable results with a simple concealment strategy, such as copying blocks from the previous frame.

In recognizing the need to provide enhanced concealment capabilities, the Video Group has developed an additional error resilient mode that further improves the ability of the decoder to localize an error.

5.4.5 Data Partition

Specifically, this approach utilizes data partitioning by separating the motion and the texture in the video packet. This approach requires that a second resynchronization marker be inserted between motion and texture information. If the texture information is lost, the motion information is used to conceal these errors. That is, due to the errors the texture information is discarded, while the motion is used to motion compensate the previous decoded VOP.

Within a typical video packet, the bit stream syntax combines together the motion and DCT data of each macro block, as depicted in Figure 5.11 .The syntax also combines the control information needed by the decoder to decode both the motion and DCT data. In this scenario, when the decoder detects an error, whether the error occurred in the motion part or the DCT part, all the data in the packet are discarded. Since the exact location where the error occurred is not known, the decoder cannot assure that either the motion or the DCT data of any of the macro blocks in the packet are not erroneous.

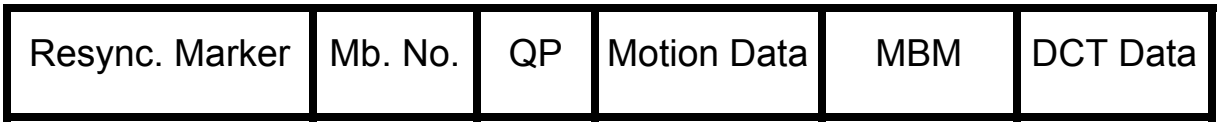

Figure 5.11: Bit stream organization with data partitioning for motion and DCT data.

The video packet in Figure 5.11 is divided as follows:

- 1st partition: Macro block number + QP + Motion Data.
- MBM: Motion Boundary Marker.
- 2nd partition: DCT Data.

Data Partitioning enables an encoder to reorganize the coded data within a video packet to reduce the impact of transmission errors. The packet is split into two partitions, the first (immediately after the video packet header) containing coding mode information for each macro block together with DC coefficients of each block (for Intra macro blocks) or motion vectors (for Inter macro blocks). The remaining data (AC coefficients and DC coefficients of Inter macro blocks) are placed in the second partition following a resynchronisation marker. The information sent in the first partition is considered to be the most important for adequate decoding of the video packet.

The Data Partitioning tool can also be exploited with the purpose of performing Unequal Error Protection. A loss of information about quantization levels and motion vectors, that is first partition, is more damaging than a loss of information about DCT coefficients, or second partition. Hence, it makes sense to give different levels of protection to the partitions according to their importance.

The use of Data Partitioning has neither impact on the encoding video quality or on the output bit rate. The influence on the encoding/decoding complexity is also negligible.

5.4.6 Fast recovery in real-time coding

5.4.6.1 NEWPRED

A newly developed technique in MPEG, called NEWPRED (for 'new prediction'), provides a fast error recovery in real-time coding applications. It uses an upstream channel from the decoder to the encoder. The encoder switches the reference frames adaptively according to the error conditions of the network. NEWPRED does not use intra refresh and it provides the high coding efficiency. This technique has been proven to work under stressful error conditions:

- Burst Error on the wireless networks (averaged bit error rate is 10E-3, 1ms burst length)
- Packet Loss on the internet (packet loss rate is 5%)

5.4.6.2 Dynamic Resolution Conversion

A special technique of use in real-time encoding situations is Dynamic Resolution Conversion (DRC), a way to stabilize the transmission buffering delay by minimizing the jitter of the amount of the coded output bits per VOP. Large frame skips are also prevented and the encoder can control the temporal resolution even in highly active scenes. This technique requires back channel information to be sent to the encoder, which explains why it is only useful in real-time situations.

6 Non-functional attributes of MPEG-4

This section analyzes the immunity of the MPEG-4 encoded sequences to network variations together with the possibility of adaptation to the varying network bandwidth constraints.

6.1 Network variation without video adaptation

6.1.1 Impact of wireless link capacity changes on video quality

Wall reflections, moving people, or other electro-magnetic interference sources easily perturb the wireless link. For example, a video is transmitted from DVD player source to the screen destination. At the transport level video frames, containing a complete picture, are transported with a bit rate that varies between 1.5 Mbit/s for a low quality digital Standard Definition TeleVision (SDTV) video and 10 Mbit/s for a high quality SDTV video. (Be aware these numbers are just indications and vary from video to video and standard to standard). At the Link layer the frames are decomposed into packets, the unit of transport over the individual links, and sent to the Access Point (AP). In the AP the packet is stored again, possibly fragmented, and sent on to the destination screen over the wireless link. The capacity of the wireless link varies between 5 to 24 Mbit/s dependent on the wireless link standard. Losses of 30% occur in the wireless link in bursts dependent on the operational conditions.

Two consequences of unwanted protocol behavior can be distinguished: (1) generation of artifacts due to packet losses (see Figure 6.1) and (2) stalling and hiccups due to late arrival of packets. The artifacts are directly connected to the structure in P, B frames and I.

Figure 6.1 Examples of blocking artefacts

The loss of an I or P frame removes the original references that are used in the B and P frames to visualize identical parts from the I or P frame. Suppose a scene change occurred, and the I-frame with the new scene has disappeared. In the B frame, parts of the former scene are then displayed on top of the new scene information. The effect is the appearance of artifacts as shown in Figure 6.1 .

Remedies against packet loss and packet corruption are done at many levels. First a checksum makes sure that the receiver rejects a corrupted packet with a high probability. Consequently at the lowest level, packet corruptions are converted into packet loss. Adding so much redundant data in the packet that corruptions can be detected and repaired can do more. Repairs usually refer to at most one bit per byte, by adding an error code at the end of each byte. It is difficult to repair bursty errors in this way. The overhead in data becomes so large that a simple resending is more cost effective in consumed bandwidth. Resending is an important aspect at the link layer of the wireless medium. Each packet arriving correctly at a specified sender (no broadcast) is acknowledged by the destination. When the sender does not receive the acknowledgement, the sender resends the packet a maximum number of times. The number of times to resend is the QoS of the link. This number lies between 1 and 256 with a favorite value around 8-16.

A frame is composed of 20-60 packets when sent over the wireless link. Loosing one packet in the total number of packets making up one frame often means a complete loss of the frame. Techniques can be employed to encode the frames such that by loosing a packet only parts of the total image is lost (e.g. a few slices). Other techniques can split the video up in important and less important information (scalable video) and make sure that the link resends the important information more often at the expense of the less important information.

The use of transport protocols aims at resending eventually lost packets lost on the path between sender and receiver. Not only the losses in the hops composing the path, but also losses inside the processors between the hops are compensated. TCP is a well-known example. However, when many packets are lost, the TCP protocol waits for a time-out to occur before resending its packets. This behavior translates itself as hiccups in the video presentation. The bandwidth availability and fluctuations determine the best transport protocol that combined with a video format, optimizes the perceived quality of the stream. We summarize the behavior of the TCP and RTP protocols for video streaming.

Assume that there is enough bandwidth for the video. With no losses there is no difference in the quality of the displayed video when transported by TCP or RTP. With few losses, TCP provides unperturbed video and RTP will show artifacts due to frame losses. With bursty losses, we must distinguish two cases: (1) video can be stopped at source (e.g. DVD player) until the sending buffer has been emptied or (2) video cannot be stopped (e.g. live football match) which leads to sender buffer overflow. With bursty losses and stoppable video, RTP will show artifacts and TCP will show freezing images during several seconds. It depends very much on exact operational conditions and user perception, which behavior is preferred. With bursty losses and broadcast, both RTP and TCP show artifacts while TCP in addition leads to frozen images. Table 6.1 summarizes the results.

	No loss	Single loss	Loss burst, broadcast	Loss burst, stoppable
TCP	Good	Good	Artifacts and freezing	Freezing
RTP	Good	Artifacts	Artifacts	Artifacts
Preferred protocol	No distinction	TCP	RTP	Undecided, depends on user perception

Table 6.1 TCP versus RTP comparison, assuming enough bandwidth

Assume there is not enough bandwidth for the video. With live broadcasts, the sender buffer will overflow, and artifacts will appear, both with TCP and RTP, With live broadcasts and losses, the TCP protocol will show freezing and artifacts. The RTP protocol will just show artifacts without freezing. However, when there are many artifacts, the freezing will not be perceived any more, and there is no difference between RTP and TCP. Suppose the video can be stopped when the send buffer is full and restarted when the buffer is empty again. Under no losses, both TCP and RTP will show the same freezing behavior. With losses, TCP will show freezing behavior without artifacts, while RTP will show artifacts as well.

We may conclude that both TCP and RTP show unacceptable video when the bandwidth is too small. Additions to both protocols are needed to make streaming over a wireless network a valid proposition. The perception of the video is dependent on the loss pattern and bandwidth fluctuations in the communication medium, the protocol on top of the medium, and the characteristic of the MPEG code that is transmitted.

6.1.2 Impact of video parameters on the network energy

6.1.2.1 Impact of the coding parameters on the network energy

In general, the higher the bit rate produced at the application layer the more energy will have to be spent at the network side to successfully transmit a higher amount of bits. Therefore, producing a lower bit rate sequence (by means of increasing the quantization parameter or reducing the spatial or temporal resolution) has as effect not only a complexity and energy reduction at the codec but also at the network. This bit rate reduction usually comes with a quality decrease.

Moreover, a high bandwidth demand coming from multiple users can cause an overload situation at the network degrading the performance an efficient utilization of resources.

6.1.2.2 Impact of the error resilience tools

6.1.2.2.1 Intra Macro block Refreshments

The use of Intra Macro block refreshments produces a bit rate increase and consequently an energy increase at the network side, unless the rate control is enabled at the encoder and the rate is kept constant while slightly degrading the quality. As these Intra MBs are some kind of redundancy so as to stop error propagation, its use in absence of errors is discouraged. Therefore, the amount of Intra Macro blocks introduced should depend on the error rate experienced at the network. As the use of Intra MB refreshes reduces the error propagation it can be used to relax the network requirements (by increasing the target Packet Error Rate (PER)). This can save then energy at the network side (reduction in transmission power, number of retransmissions) while increasing the energy at the encoding side. The optimal tradeoff in terms of effort spent at the application or at the network still needs to be explored.

6.1.2.2.2 Resynchronization Video Packet

The video packet size can have a big impact on the processing resources at the network side. If each video packets is packetized in a separate RTP/UDP/IP packet, (so as to maintain the decoding independency) which in turns is encapsulated in a MAC (Medium Access Control layer) packet then the use of small packet sizes at the application level implies a bigger overhead at the network due to the higher number of packet headers. In addition, the added overhead at the MAC layer is highly dependent on the packet size but also on the physical layer modes of the MAC 802.11a as Figure 6.2 shows.

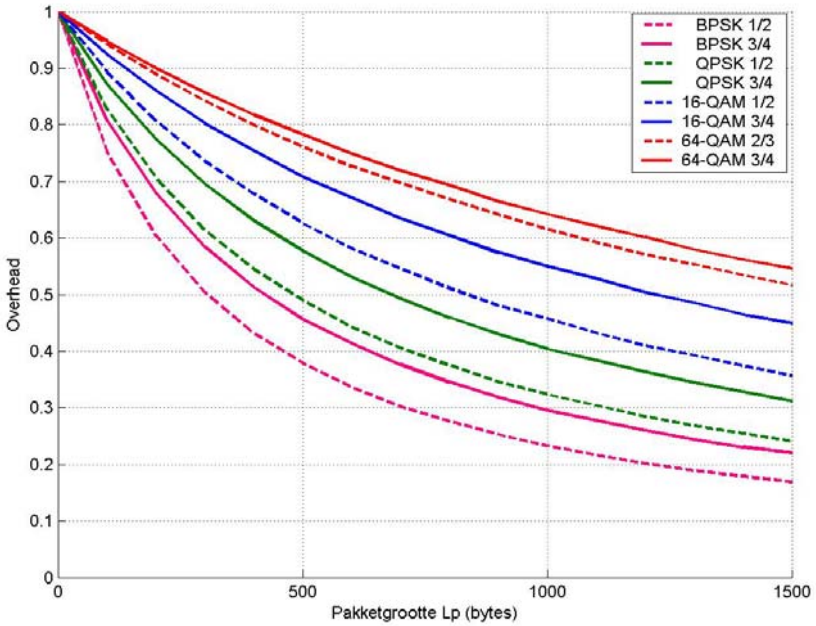

Figure 6.2: Overhead of MAC 802.11a DCF.

In practice, the selection of a packet size is a tradeoff between the resilience provided and the overhead caused mainly at transport and MAC layers.

6.1.2.2.3 Data Partition

As stated in the previous chapter, the Data Partitioning tool is exploited with the purpose of performing Unequal Error Protection (UEP). Moreover the use of data partition is not efficient if there are no means to perform UEP. The loss of different kind of information has a different impact on the end quality. Hence, it makes sense to give different levels of protection to the partitions according to their importance. This can increase the end quality by allowing a limited awareness of the video content at the network. At the same time energy can be saved by optimally allocating the protection among the data and reducing the amount of protection on less relevant data.

6.1.2.2.4 Error Concealment

Finally, the use of error concealment also alleviates the transmission errors. A better concealment technique at the decoder increases on one hand the decoding complexity is increased but on the other hand the network requirements can be relaxed (in terms of target PER and therefore also in energy cost) as the video sequence can recover better from transmission errors.

6.2 Network variation with video adaptation

To successfully maximize the end video quality under bandwidth and processing power constraints requires video adaptation to the varying network conditions.

6.2.1 Adaptation of Coding Tools

Video adaptation to varying bandwidth (or similarly to varying processing constraints) can be achieved by tuning the video parameters to scale the encoder output bit rate. This is done by tuning the Quantisation Parameter (SNR scaling), or by reducing the temporal or spatial resolution (temporal or spatial scaling). When the channel conditions get worse it can be advantageous that the codec reduces its output bit rate (reducing then the encoded video quality), which may help the network to succeed in transmitting a lower amount of bits but with less transmission errors. This way, the encoded or error-free quality decreases (less visually annoying) while the end quality increases if the transmission errors (visually more annoying) are reduced.

6.2.2 Adaptation of Error Resilience Tools

The error resilience tools can also be adapted to the varying channel conditions. This way, the percentage of Intra MBs depends on the packet error rates that the wireless link is suffering. If there are few transmission errors there is no need for adding extra redundancy (Intra MBs are less compressed) at the video codec as this overprotection will involve a loss of coding efficiency. In presence of higher errors a higher percentage of Intra MBs may be needed to combat the error propagation.

In a similar way the size of resynchronization packet needs to adapt to channel conditions. Larger packet sizes (coinciding with MAC frames) involve less overhead but on the other hand are more prone to be hit by errors. Therefore, under worse channel conditions it may be necessary to reduce the packet size to tradeoff some extra overhead by a higher robustness against errors.

7 Conclusions

This document provides an inventory of MPEG-4 codecs so as to, based on the codec characteristics and scenarios under consideration in BETSY, propose suitable MPEG codecs.

In the scenarios described in the BETSY project the need of scalability cannot be motivated by the existence of multiple heterogeneous devices receiving a common content. However, adaptability to varying bandwidth and processing requirements is needed in the codec to avoid rapid quality degradation. Energy efficiency becomes an important issue as well as we are dealing with energy constrained mobile devices. This adaptability can be provided by means of tuning the coding parameters of a single layer codec.

Previous analysis of the available codecs at its multiple profiles, we can conclude that for the scenarios under consideration in BETSY the MPEG-4 Part 2 Simple Profile codec is a good choice as it offers a good tradeoff between coding efficiency, error resilience and implementation complexity. It provides as well the adaptation capabilities required for our scenarios. If the need for higher coding efficiency and resilience capacities justifies a complexity increase, a simple configuration of the Advanced Video Codec (MPEG-4 Part 10) may be taken under consideration.

References

[1] Iain E.G.Richardson, “H.264 and MPEG-4 Video Compression”, Wiley 2003.

[2] Weiping Li, Jens-Rainer Ohm, Mihaela van der Schaar, Hong Jiang, Shipeng Li, “MPEG-4 Video Verification Model version 18.0”, ISO/IEC JTC1/SC29/WG11 N3908 Pisa, January 2001.

[3] G. Sullivan et al., “Rate-distortion optimization for video compression”, IEEE Sign. Proc. Mag., 1998 15 (6) pp. 74-90

[4] D. Marpe, G. Blattermann, G. Heising, T. Wiegand, “Video compression using context-based adaptive arithmetic coding”, Proc. IEEE ICIP 2001, pp. 558-561, Thessaloniki, Greece, October 2001

[5] ITU-T and ISO/IEC JTC1, “Advanced Video Coding for Generic Audiovisual Services,” ITU-T Recommendation H.264 – ISO/IEC 14496-10 AVC, 2003.

[6] ITU-T and ISO/IEC JTC1, “Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods”, JVT-N046, Jan 2005

[7] W. Sweldens, “A custom-design construction of biorthogonal wavelets,” J. Appl. Comp. Harm. Anal. vol. 3 (no. 2), pp. 186-200, 1996.

[8] ITU-T and ISO/IEC JTC1, “Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video,” ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2), 1994.

[9] ITU-T and ISO/IEC JTC1, “JSVM 0 Software”, JVT-N22, Jan 2005.

[10] ITU-T and ISO/IEC JTC1, “Scalable Video Coding - Working Draft 1”, JVT-N020, Jan 2005

[11] ISO/IEC JTC1, “Verification Model 18.0 of MPEG-4 Visual,” ISO/IEC JTC1/WG11 Doc. N3908, Feb 2001.

[12] K. Denolf, et al., “Cost-efficient C-level design of an MPEG-4 video decoder”, Proceedings of the IEEE Workshop on Power and Timing Modeling, Optimization and Simulation, pp.233-242, Goettingen, Germany, September 2000.

[13] K. Denolf, et al., “Memory centric design of an MPEG-4 video encoder”, to appear in IEEE Trans. Circuits and Systems for Video Technology, special issue on Integrated Multimedia Platforms.

[14] H.M. Radha, M. van der Schaar, and Y. Chen, “The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP”, IEEE Trans. on Multimedia, vol. 3, No. 1, March 2001.

[15] Zhou Wang, Ligang Lu and Alan C. Bovik, “Video Quality Assessment Based on Structural Distortion Measurement” in *Signal Processing: Image Communication*, vol. 19, no. 1, January 2004.

[16] Zhou Wang, Alan C. Bovik and al, “Image Quality Assessment: From Error Visibility to Structural Similarity” in *IEEE Transactions on Image Processing*, vol. 13, no. 4, April 2004

[17] Kristof Denolf, Christophe de Vleeschouwer, Gauthier Lafruit and Jan Bormans, “Scaling Bandwidth and Complexity of Hybrid MC/DCT Video Codecs” in Packet Video Conference, April 2002

[18] C. De Vleeschouwer and T. Nilsson, “Motion estimation for low power video devices”, Proceedings of the IEEE International Conference on Image Processing (ICIP01), Thessaloniki, Greece, October 2001, pp. 953-957.

[19] F. Catthoor, et al., “Custom Memory Management Methodology”, ISBN 0-7923-8288-9, Kluwer Academic Pub., 1998.

[20] C. Calafate and M. Malumbres, “Testing the h.264 error-resilience on wireless ad-hoc networks," in Proc. EURASIP Video/Image Processing and Multimedia Communications Conference (VIPMCC), pp. 789-796, 2003.

[21] M. Walker and M. Nilsson, “A study of the efficiency of layered video coding using H.263”, PacketVideo99, New York, April 1999.

[22] M. Budagavi, W.R. Heinzelman, et al., “Wireless MPEG-4 Video Communication on DSP Chips”, IEEE Signal Processing Magazine, January 2000, pp. 36-53.

[23] G. Cote and F. Kossentini, “Optimal intra coding for blocks for robust videocommunications over the internet,” *Image Communication* 1, pp. 25-34, Sept. 1999.

[24] Kristof Denolf and Carolina Blanch, “Initial Memory Complexity Analysis of the JVT Codec”, ISO/IEC JTC1/SC29/WG11 MPEG02/M8028 Jeju, March 2002.

[25] Sergio Saponara, Carolina Blanch, Kristof Denolf, Jan Bormans, “The JVT Advanced Video Coding Standard: Complexity and Performance Analysis on a Tool-by-tool basis”,in Packet Video Conference, 2003.

[26] Jens-Rainer Ohm. Bildsignalverarbeitung fuer multimedia-systeme. Skript, 1999.