H.264 Source for Mobile Video Streaming

Lorenzo Rossi

Dottorato di Ricerca in Ingegneria dell'Informazione e della Comunicazione
Universitá degli Studi di Roma "La Sapienza"

Dipartimento INFOCOM

A thesis submitted for the degree of

Doctor of Philosophy

Imagination is more important than knowledge
A. Einstein

Alla mia mamma

Acknowledgements

I owe my deepest gratitude to my advisors, Profs. Scarano, Cusani, and Colonnese, for their thougful guidance throughout my Ph.D.

Many thanks to my colleagues, whom I shared with the good and sometimes the bad times of my staying at the university.

To the people at EPFL, who helped me in my ”swiss” experience.

I am grateful to all my friends and parents for their loving support.

A very special thank is to my girlfriend Fiorella, who supported and encouraged me in the difficulties.

Introduction

In the recent years a lot of effort has been spent in developing new technologies for the video communications. As that electronic tecnology is advancing in producing cheap, low-cost Digital Signal Processors (DSPs), new solutions in telecommunications became possible. Few examples include new standards in video coding, from MPEG-1 (39) in the early nineties to the state-of-the-art H.264 (65) that, in its latest extension, support 3D and multiview video; the availability of large bandwith networks, such as xDSL or WiMax or 3G cellular networks. Such innovations allow the raise of new services unexpected as far as ten-fifteen years ago. Some examples are:

- video call;
- video conference;
- high speed internet connections;
- streaming services.

Most of the services are available both in fixed networks, and, in the recent few years, in mobile networks.

This work is devoted to the study and the analysis of the problematics regarding a mobile video streaming system employing the novel video standard ITU-T Rec. H.264, also known as ISO/IEC MPEG-4/Part 10-AVC (65). Mobile streaming services are arising in the last few years thanks to the evolutions in the mobile networks, such as 3G (UMTS), 3.5G and wifi networks. These networks suffer of variable channel characteristics, due to the nature of the mobile/wireless medium. The variability of the channel results in a variable bandwith with packet

losses and delays. From the point of view of the video transmission this is highly undesiderable, because only a single packet loss may results in several missed or distorted frames.

This thesis explores some of the problematics regarding the trasmission and playout of video content in a mobile environment. At first, after the introduction of the main themes, the thesis deepens the topic of video rate control, *i.e.* techniques employed at the encoder in order to have bitstreams with costant average bit rate and uniform quality. A technique for selecting the proper kind of frame is presented. The second topic is on error resilience/concealment techniques, the former devoted to encode video so that, after packet losses, video can be recovered, the latter devoted to hide errors due to losses in the transmission. The contribution is developing a novel scheme based on Multiple Description Coding (MDC) that introduces a reliability measure to improve image quality after the decoding stage. Third topic regards traffic models for a video streaming source; the study focus on a source that performs bitstream switching using H.264 SP frames; a process member of the Hidden Markov Model (HMM) family is introduced to efficiently describe the statistical characteristics of the video source. Last topic pertains the novel extension of H.264, Multiview Video Content (MVC), devoted to encode efficiently multiview video, a new video content that is arising recently. A traffic model for MVC source is also introduced.

The thesis is organized as follows: Chapter 1 introduces the main topics of the work, Chapter 2 illustrate the problem of rate control, Chapter 3 regards techniques of error resilience/concealment, whereas Chapter 4 describes a traffic model for video streaming system. Finally, Chapter 5 is on Multiview Video Content (MVC). Chapter 6 concludes the thesis.

Contents

Chapter 1

Introduction

This chapter introduces how a video streaming system is composed and its characteristics and issues. In Section 1.1 a brief description of the H.264 standard is exposed, and in Section 1.2 a video streaming system is introduced. Sections 1.3, 1.4, 1.5 and 1.6 expose the main themes of the work. Section 1.7 concludes the chapter.

1.1 H.264/AVC

The state-of-the-art of video coding ITU-T Rec. H.264, also known as ISO/IEC MPEG-4/Part 10-AVC (65) has been defined by the joint effort of two groups, the ISO MPEG group and the ITU VCEG group, that formed the Joint Video Team (JVT) in 2001. The first version of H.264 was completed in 2003, from then some extensions have been released, most notably the Scalable Video Coding (SVC) in 2007 and the Multiview Video Coding (MVC) in 2009.

H.264 is used in such applications as players for Blu-ray Discs, videos streaming services, broadcast services for DVB and SBTVD, direct-broadcast satellite television services, cable television services, HDTV and real-time videoconferencing. This wide application field is guaranteed by the several novel technical enhancemente introduced by H.264:

- intra spatial prediction;
- high freedom in motocompensation;

- integer DCT-like transform;
- Context-Adaptive Binary Arithmetic Coding (CABAC) to efficiently compressy data;
- definition of new kinds of frame for streaming applications.

The standard is separated in two layers: the first one, named Video Coding Layer (VCL) is devoted to the encoding and decoding; whereas the second layer, named Network Adaptation Layer (NAL) describes how to adapt the bitstream to existing networks. In the next subsections we describe the characteristics above.

1.1.1 Video coding layer

In this section we describe the main innovations of H.264 regarding compression efficiency.

1.1.1.1 Intra spatial prediction

H.264 is the first standard in video compression introducing a prediction in the spatial domain by estimating a block from the surrounding pixels. The block may be of different sizes: 4x4, 16x16. Large blocks are useful in predict smooth areas of the image. The 4x4 blocks support four types of prediction: in Fig.1.1 are shown 5 types. The 16x16 blocks support 4 kind of prediction.

1.1.1.2 DCT-integer transform

One of the drawbacks of the DCT transform is the infinite precision required to represent the content with no errors on different machines. H.264 uses a different transform, based on DCT, that works only with integer numbers, avoiding completely that issue. The transform works on 4x4 block, since different kind of predictions (intra or inter) are employed in 4x4 blocks. In addiction, H.264 includes other two transforms, Hadamard transform for 2x2 and 4x4 blocks, to efficiently encode DC coefficients of chromas and luma in a macroblock.

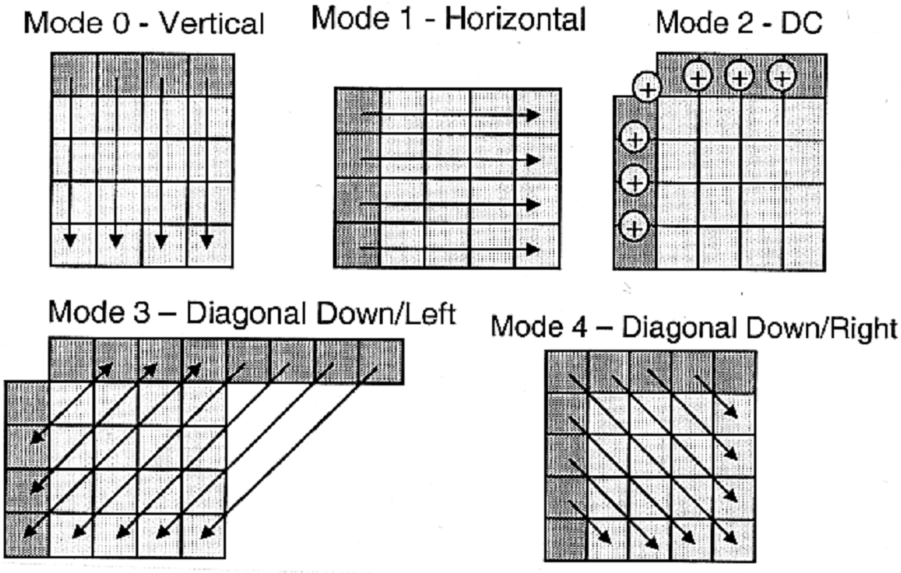

Figure 1.1: Five of the nine 4x4 Intra prediction mode (63).

Quantization is similar to previous standard, except for the relationship between QP and Q_{step}:

$$Q_{step} = 2^{(QP/6)}$$

due to the some simplification involved in the DCT-integer transform. Fig.1.2 shows the difference between H.264 and H.263 on defining QP.

1.1.1.3 S frames

Switching frames are first introduced in H.264 (31). The S frames are divided in two categories: SP frames (primaries and secondaries) and SI frames. Both are used in the following tasks:

- bitstream switching;
- splicing and random access;
- error recovery;
- error resiliance.

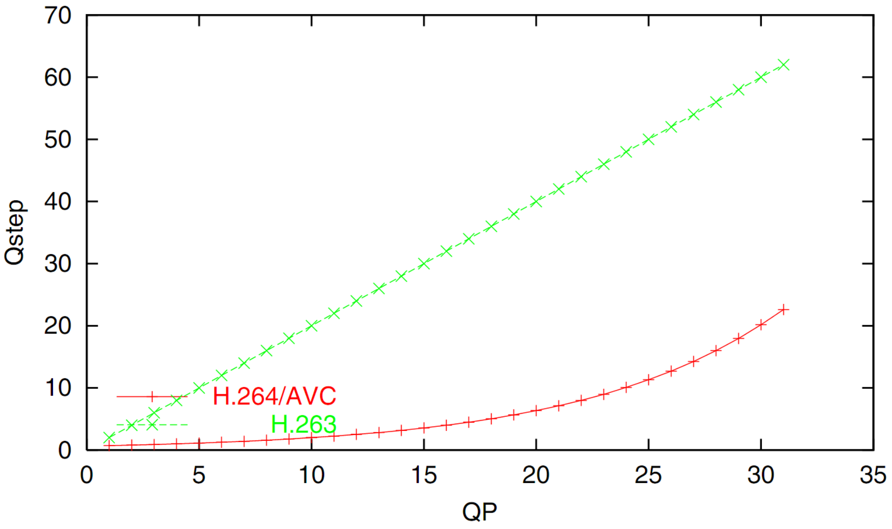

Figure 1.2: The relation between QP and Q_{step} (43).

The first item is discussed deeply in Section 1.2, here we briefly describe their behavior. The main feature of SP frames is that identical SP frames can be reconstructed even when different reference frames are used for their prediction. A primary SP frame can replace a P frame in a bitstream and the encoder can generate a secondary SP frame or a SI frame in corrispondence to the primary one. The SI frame (I is for Intra) is an intra frame that, once decoded, is identical to the primary SP frame. The secondary SP frame is predicted from different frames of the primary one but it results identical, pixel per pixel, to the primary one, once decoded, thanks to the motocompensation performed on the transformed domain instead on the spatial domain. Two quantization parameters are employed for selecting the better compromise between the sizes of the three frames. In Fig.1.5 the schemes for, respectively, decoding of secondary SP/SI frames, decoding of primary SP frames, encoding of SP frames are shown.

The price for the flexibility is in the lower rate-distortion curve with respect to classical I,P frames (50): primary SP costs more than a P frame, whereas secondary SP and SI frames are comparable to I frames. The gain consists in transmitting primary SP frames more frequently than secondary SP and SI frames; previous encoding schemes used I frames which cost is larger than primary SP.

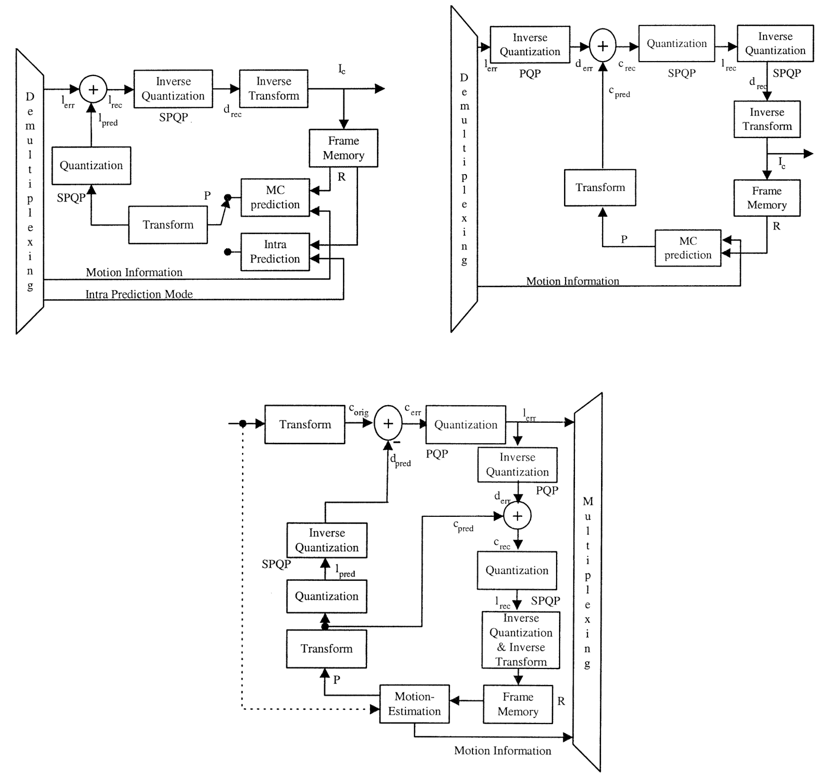

Figure 1.3: Encoding and decoding schemes for S frames (31).

1.2 Video streaming system

In this section we introduce a modern video streaming system, its application fields, the main characteristics and issues.

From a historical point of view, video transmission has been first encountered in the framework of broadcasting and conversational services. Broadcasting is characterized by large available bandwidth and medium-to-high required quality, with loose coding delay requirements. On the contrary, conversational services are characterized by significantly reduced bandwidth and quality, with severe coding delay requirements.

Recently, video streaming is emerging as a promising application over fixed, wireless, and heterogeneous networks. Video streaming services are characterized by contemporary download and presentation of the video content, with short initial buffering delay. The typical architecture for video streaming consists of

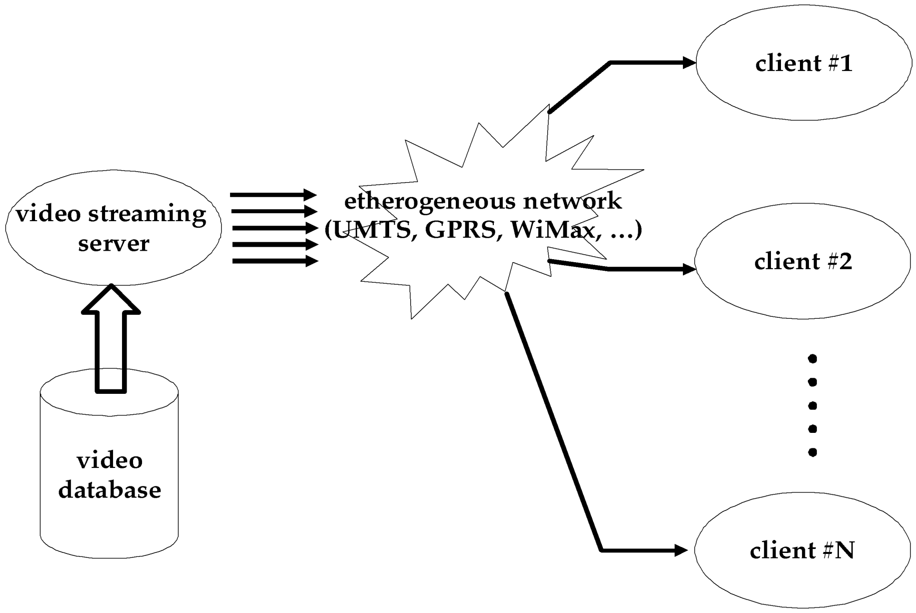

Figure 1.4: A video streaming system.

a server that transmits data extracted from pre-encoded sequences to users that have requested a video content. Fig.1.4 shows a typical video system architecture.

The pre-encoded sequences are typically stored along with meta-data, such as packetization instructions or session description information (1), which are exploited during the on-going streaming session. The peculiar requirements of streaming services reflect into video traffic characteristics. First, video traffic does not require a strict rate-control as for conversational services over circuit-switched or packet-switched networks; rather, loose rate control mechanism are envisaged, mainly attempting to comply with system requirements such as peak and average bandwidth, decoding buffer size and decoder complexity. Second, streaming takes place on channels that may vary due to physical channel changes, to vertical handovers caused by user mobility among heterogeneous networks or even to dynamical network re-configuration, as in Wireless Mesh Networks. All these causes result into variations of the available average user bandwidth, causing the server to react by adapting the rate of the transmitted video so as to accommodate these variations.

In streaming services, rate adaptation should be achieved without encoding the sequences in real-time using different coding parameters, as it occurs in conversational services, but rather accessing to suitably previously encoded video streams of different bandwidth and quality. Thus, the server should be able to perform dynamic switching between previously encoded bitstreams to face variations of the bandwidth available to each user. At this aim, the encoded sequence should provide random access point for bitstream switching, in a fashion resembling that of Intra coded pictures in broadcasting services. This issue has been explicitly accounted for in H.264. In fact SP frames, which enable perfect frame reconstruction even using different reference frames for motion compensation (31) can be exploited in video streaming applications during bitstream switching since they provide random access points to the bitstreams, at a lower coding cost than classical Intra coded pictures.

We now describe how a video streaming server employing H.264 SP frame works. Let us assume that the video sequences to be transmitted are encoded in L different bitstreams, each one at different average bit-rates r_i, $i = 1, \ldots, L$. The server selects the proper sequence according to users' feedback and network conditions. The H.264 encoder places SP frames in each stream, so to provide virtual access points in order to enable bitstream switching. Typically, the SP frames are periodically inserted, so that each SP frame is followed by a fixed number $N_{\mathrm{GOP}} - 1$ of encoded non-switching frames; then, the encoded bitstream is constituted by GOPs beginning with an SP frame, in a fashion resembling the MPEG-2 GOP structure beginning with an Intra coded frame. Each random access frame is encoded by means of the so-called primary SP representation. Besides, also different representations, the so-called secondary SP representations, are encoded using reference frames belonging to different bitstreams for motion compensation. Thanks to the SP coding syntax, the coded primary and secondary representations result in exactly the same decoded frame. During the streaming, the SP primary representation is sent to users that are continuously decoding the i-th bitstream; the SP secondary representation is only needed by users that performs switching from a different bitstream, and therefore start decoding the i-th one using a different decoded sequence for motion compensation. The picture decoded using the primary and the secondary SP frame is just the same, and the bitstream switching is drift free; however, since motion compensation is used both for the primary and secondary SP frame, the encoding cost of a random access frame is low compared with the cost that would have been required by periodic insertion of Intra frames. Fig.1.5 illustrates an example of how bitstream switching can be realized using the H.264 SP syntactic structures during a video streaming session. With reference to the notations adopted in Fig.1.5, frame D is decoded during continuous streaming of the same bitstream, if the data labeled as {A, B, C, D (Primary SP)} are streamed, as well as after a bitstream switching, if the data labeled as {1, 2, 3, 3-D (Secondary SP)} are streamed.

During continuous streaming of the i-th video bitstream at average rate r_i, the emitted GOPs begin with a primary SP; when switching is needed between the i-th video bitstream at average rate r_i and the j-th video bitstream at a different average rate r_j, a suitably coded secondary SP is sent, followed by $N_{\text{GOP}} - 1$ non-switching frames. Then, we can classify GOPs into various classes, differing in the kind of SP frame they begin with, and in the average rate at which they are transmitted. Specifically, we identify L^2 classes of GOPs, namely L classes of GOP beginning with primary SP extracted from a stream at average rate $r_i, i = 1, \ldots, L$, and $L \cdot (L-1)$ classes of GOP beginning with secondary SP, transmitted when a switching between the rates r_i and r_j, with $i, j = 1, \ldots, L$ and $j \neq i$ occurs. GOPs emitted by a real video source during video streaming satisfy precise decoding constraints. In particular, switching between different rates is realized by transmitting a GOP beginning with a secondary SP, since this latter enables the decoder to perform motion compensation using the already received frames without error-drift. Hence, GOP structures of different classes are not generated in arbitrary order, but the class of each transmitted GOP depends on the class of the previously generated GOP.

As exposed above, a video streaming service has started to be employed in wireless networks, as 3G or 2.5G networks; since those networks exhibit non-stationary behavior and provide a smaller bandwidth than fixed networks, several issues arise. The aim of this work consists in analysing and solving those issues.

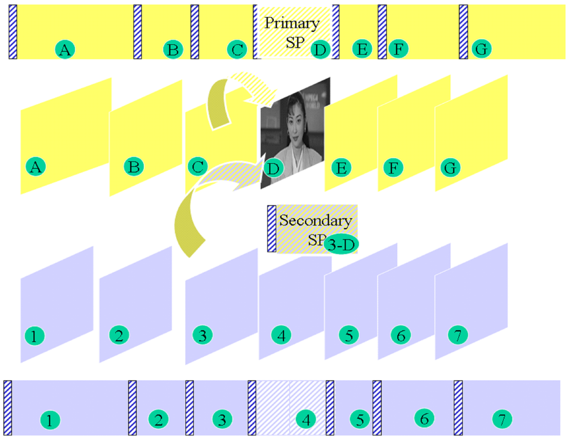

Figure 1.5: Example of H.264 video streaming session: frame D is decoded during continuous streaming of the same bitstream, if the data labeled as {A, B, C, D (Primary SP)} are streamed, as well as after a bitstream switching, if the data labeled as {1, 2, 3, 3-D (Secondary SP)} are streamed.

1.3 Improving compression efficiency

In the phase of encoding the bitstream several decisions can be made in order to maximize the visual quality of the whole video content still respecting the constraints imposed by the system. In fact, in bitstream applications, loose rate control is applied to the single streams in order to maintain an average bit-rate without exceeding too much. Coversely, streaming applications with no adaption of the stream to channel conditions use severe rate control techniques in order to maintain a Costant Bit Rate (CBR) behavior. Under these constraints the encoder can choose how allocate the budget (in terms of bits) among the frames in a GOP or even among the macroblocks inside a single frame. Moreover, some sort of flexibility can be allowed in selecting the kind of frame (macroblock), *i.e.* if a frame is I, P, B, SP. Of course the best solution is using only B frame that exhibit the best rate distortion efficiency, but in a streaming context, in case of frame losses (not rare events), such decision may cause a severe degradation of the visual quality. A typical rate control procedure follows the steps:

1. The bit budget for the GOP is divided among the frames according to a rule depending on the innovation of the frames;

2. the bits are divided between macroblocks according to the visual relevance;

3. A quantization parameter is chosen for each macroblock in order to have the bit size selected in step 2.

Typical measure for evaluating visual relevance are: Mean Square Error (MSE), Sum of Absolute Difference (SAD), Mean Absolute Difference (MAD):

$$\begin{aligned}MSE :&\frac{1}{MN}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}(x[m,n]-\hat{x}[m,n])^2\\ SAD :&\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}|x[m,n]-\hat{x}[m,n]|\\ MAD :&SAD/MN\end{aligned} \tag{1.1}$$

where $x[m,n]$ is a frame (macroblock) and $\hat{x}[m,n]$ is its estimation, depending on the prediction (intra, inter, which kind of inter). M and N are the dimensions of the frame (macroblock).he

The rate control for H.264 is more difficult than those for other standards. This is because the quantization parameters are used in both rate control algorithm and rate distortion optimization (RDO) (64), resulting in the following chicken and egg dilemma: to perform RDO for macroblocks (MBs) in the current frame, a quantization parameter should be first determined for each MB by using the mean absolute difference (MAD) of current frame or MB. Several techniques have been developed to move around it (2).

1.4 Improving error resiliance

Video content is transmitted in packets, usually fixed-size ATM cells or variable length RTP/UDP datagrams through the network. The tranmission may suffer of error and losses: in fact video streaming in wireless networks may suffer of difficult conditions due to the environment, resulting in packet losses (absence of signal), several packet errors, large delays, etc. Moreover, buffer overflows and underflows can cause other losses. Since error correcting techniques are not always capable to recover packets, some data might not arrive to the video decoder, causing a severe degradation of the video, because of the dependency of the encoded video data (motocompensation,etc...).

Many techniques have been developed in order to mitigate the degradation due to the losses. These techniques are divided in two main categories: error resiliance and error concealment. The former includes strategies at the encoder stage developed to fortify data in case of losses, *e.g.* sending two versions of the same frame, partioning data in more packets, adding redundancy to data, etc, the latter includes strategies at the decoder to hide errors, *e.g.* in case of frame loss, replacing it with the last frame available. More sofisticates strategies include estimation and prediction of loss data.

H.264 introduces some data structures in order to manage easily with the error resiliance and concealment. The following structures are introduced:

- slice structure;
- random intra refresh (RIR);
- data partitioning;
- flexible macroblock ordering (FMO);
- arbitrary slice ordering;
- redundant slices.

The definition of slice is the only one defined in previous standard (H.263, MPEG-4): a slice is a segment of frame usually of fixed size in terms of macroblock, or rarely, in bytes. Each frame is divided in a fixed, integer number of slice that are encoded independently so as, in case of slice loss, the other slice can be still decoded. Usually each slice is sent in a single packet. The RIR is an encoding strategy that forces a fixed number of macroblocks in each frame to be intra encoded, so, in case of losses, the video content can be recovered quickly. Macroblock selection is random; if the selection is not random but depends on the video content the strategy is called Adaptive Intra Refresh (AIR). Data partitioning consists in dividing slice data in three sets (partitions) according to the relevance with respect to the decoding process. Partition A contains header, motion vectors, macroblocks prediction modes. Partition B contains intra coefficients and inter coefficients are in partition C. Data partitioning allows

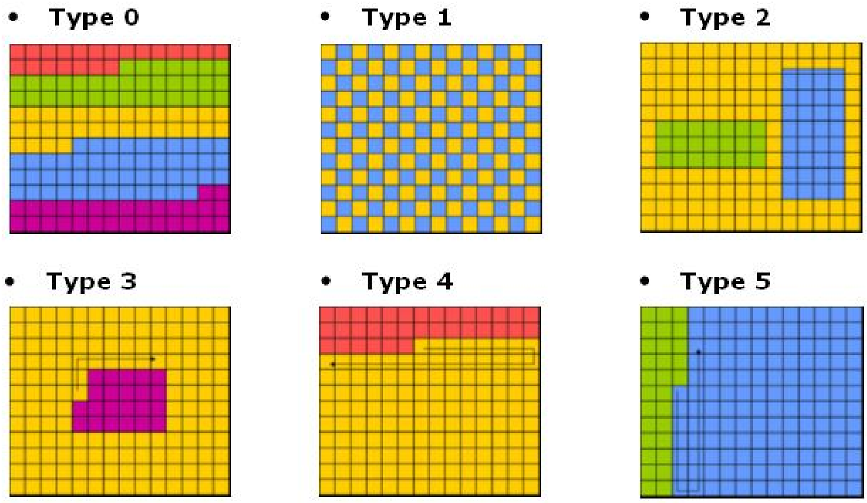

Figure 1.6: Different kind of flexible macroblock ordering. Type #1 is used for error resiliance.

to differentiate the resiliance strategies depending on the importance of the partition. The arbitrary slice ordering is used to interleave data at the video content level, making video transmission robust to burst errors. Of course, as in typical interleaving, the decoder have to wait the arrival of all the slices of a frame. FMO allows to build slices using not adjacent macroblocks; in case of slice loss, the macroblocks are spreaded in the whole frame and efficient error concealment strategies can be employed to recover the frame. Fig.1.6 shows different kind of FMO. The last one, redundant slice, is used to transmit a coarse version of a slice. In case of loss the decoder can use the redundant slice to recover missing data.

The drawback of the resiliance strategies is in a larger data to be transmitted: in fact each strategy adds redundancy (in a different way) to the video content.

1.5 Video traffic modeling for streaming application

The phase of network dimensioning is crucial for the system, in fact a wrong dimensioning of the network resources can severely degrade the communication,

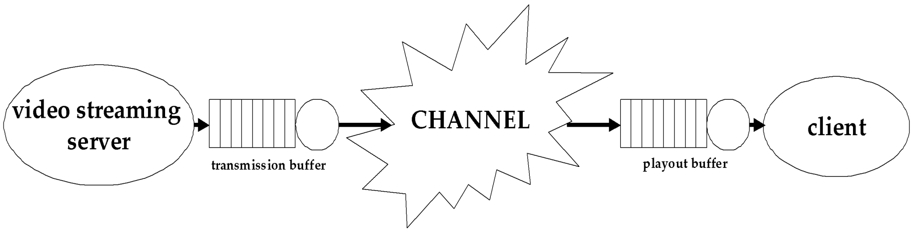

Figure 1.7: Video streaming system with buffers.

or conversely, employ too many resource in the network. One of the most frequent cases, is buffers size.In a video streaming system, as other systems, buffers are employed for traffic shaping, de-jittering and other purposes. Fig.1.7 shows a typical video streaming system with two buffer, the first one to shape traffic at the channel entrance, the the second to decouple the network from the client and provide data to the decoder at costant rate. Sometimes two buffers are used to separate the two functionalities. In network resource allocation, buffer dimensioning is really crucial, because a buffer overflow may occur resulting in packet losses. In order to design in a proper way the buffers size, an accurate analysis of the data traffic generated by the video encoder should be performed. A useful tool for helping the network project manager is a statistical model of the video source. In literature several studies pertain to analyse and characterize video traffic, in the context of broadcast application or teleconferences. The most straightforward application of a video source model consists in generating a pseudo-random syntetic sequence, used for dimensioning network resources, so there is no need to encode real video.

In Chapter 4 we present a video streaming traffic that efficiently mimic the behavior of a real streaming source.

1.6 Improving user experience: Multiview Video Coding

The final step of the work is the analysis of the newest extension of H.264, the Multiview Video Coding (MVC). In this section we introduce the topic and describe the main characteristics of MVC. In Chapter 5 a MVC video traffic model is presented.

1.6.1 Introduction

In the recent years, due to the recent advances in technology, a number of new applications involving the combination (*e.g.* 3D videos) of two of more video sequences are becoming to be accesible to the consumer. These applications include 3DTV, free-viewpoint applications, just to mention few examples. Although the existing video coding standards (MPEG-4, H.264, etc.) may be employed to encode them, a suitable standard turned entirely to encode and transmit these applications is high desirable. The Joint Video Team (JVT) is developing an extension of the widely spread video coding standard H.264/AVC in order to define a new encoding technique able to efficiently encoding and transmit the video streams. Such extension is named Multiview Video Coding (MVC, H.264/MVC). The chapter is organized as follows: in Sec.1.6.2 we describe the typical application in which the MVC may be employed; in Sec.1.6.3 MVC is introduced. Sec. 5.4 concludes the chapter.

1.6.2 Applications

A typical MVC scenario consists in a scene recorded by several cameras with a different angle, see Fig.1.8. The streams are called *views*, and are trasmitted to the users.

The possible applications involving a multiview scenario can be grouped in three main classes: free-viewpoint video, 3D TV, immersive teleconferencing (8). We describe all these classes below.

1.6.2.1 Free-viewpoint video

The free-viewpoint video system allows the user to select an arbitrary point of view for watching the video content. The selection of the point of view may be static (determined only at the beginning of the transmission) or dynamic (the user can switch among the points of views during the transmission). If a selected point

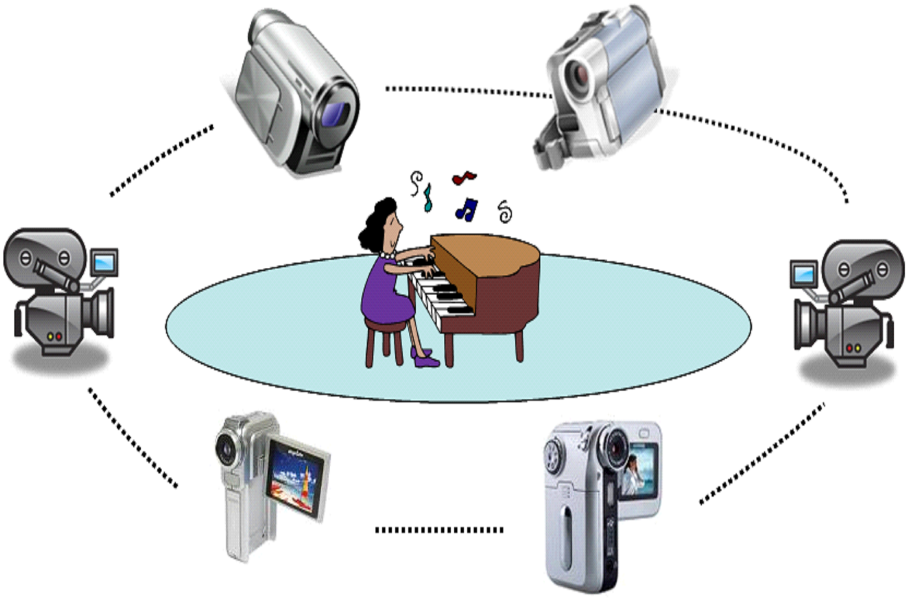

Figure 1.8: Example of Multiview Scenario.

of view is not correspondent to a camera, more views are transmitted in order to generate such point through suitable interpolation. A typical architecture of this scenario consists in a server that transmits the view(s) requested by the user; the set-top box will perform the interpolation. Other two possible architectures differ in which views are transmitted to the user: i) the server transmits *always* all the views and the set-top box selects only the views needed for the interpolation and discard the others; ii) interpolation is performed at the server side that transmits only the sequence actually watched by the user.

1.6.2.2 Interactive TV

A particolar case of free-viewpoint video is the interactive TV; the user can select only one of the views per time, interpolation is not performed. The server can transmit only requested views or all the views letting the selection to the set-top box.

1.6.2.3 3D TV

The 3D TV expands the concept of television in a 3D environment. The users, using particular glasses or other means, have the feeling of watching a real 3D scene. A minimum number of two views is needed to generate the 3D effect. In this scenario the server transmits all the views to the users that, through particular tv equipment, are able to generate 3D content. A combination of 3D TV with free-viewpoint video is possible: the users select (statically or dinamically) a point of view to watch a 3D scene. The system architecture is a like the one in free-viewpoint video.

1.6.2.4 Immersive teleconferencing

In a typical teleconference the users can see only a narrow area around the interlocutors. Immersive teleconferencing, using more cameras, are able to show a wider area.

1.6.3 Standard

Multiview Video Coding (MVC) is an extension of the widely-known video coding standard H.264. Its aim is in defining a video standard for new emerging applications, such as free-viewpoint video, 3DTV, immersive teleconferencing, interactive TV. MVC involves the encoding of several video sequences (views) representing the same scene recorded simultaneously from multiple cameras. The user, depending of the particular application may request either one view, either more than one.

Since the data to be transmitted/stored is much larger than in the usual single-view coding (SVC), the design of MVC has taken into account as primary issue a high compression efficiency, achieved by exploiting the correlations among the views: in MVC pictures from different view can be used as reference pictures for the frame to be encoded. This kind of prediction is named interview prediction, an example is given in Fig.5.1 in which the arrows denote the relations of dependency among the pictures. Let us remark that the interview dependency is only among pictures of the same istant. Letters I, P, B denote classic Intra, Predictive, and

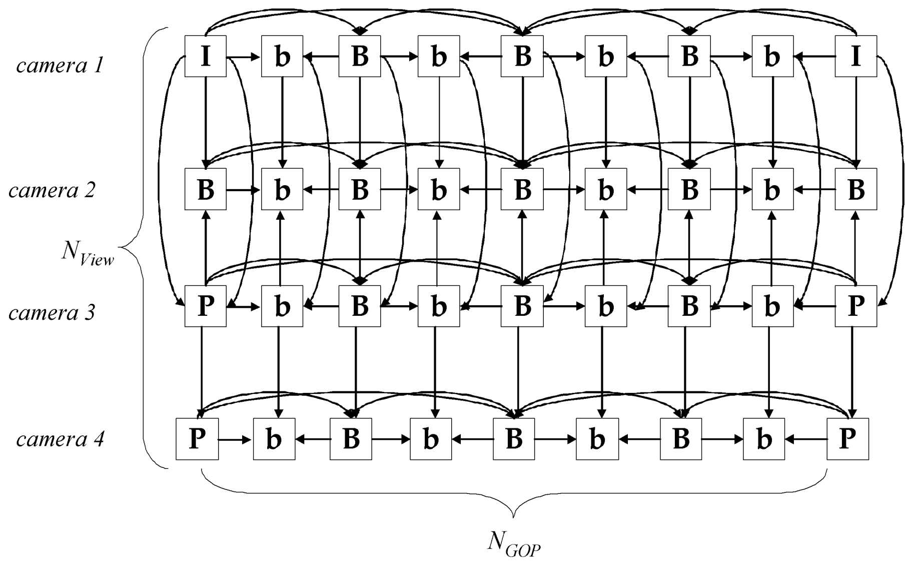

Figure 1.9: Typical GOP structure in MVC. I represents Intra Frame, P represents Predicted frames, B and b represent Bi-directional predicted frames.

Bi-directional predictive frames; the b letter denotes also Bi-directional predictive pictures with the lower case remarking that in case of temporal scalability thay are the first frames to be dropped (see below). The drawback of this strategy is in a large con consumption of resources, such as processing, buffers size, amount of data to be transmitted, etc.

The following aspects have been taken into account during the phase of MVC standardization:

- Scalability
- Decoder complexity
- Parallel processing
- Random Access
- Robustness

In the following we describe i) the structure of H.264/MVC bitstream and its innovation with respect to H.264/AVC; ii) how sub-bitstreams can be extracted from a main MVC stream; iii) decoding ordering of the stream; iv) parallel encoding of the views.

1.6.3.1 MVC bitstream

One of the design principles of MVC is the backward compatibility with H.264/AVC: one of the views is encoded independently from the others, see Fig.5.1, camera 1. The bitstream is organized so as to a compliant H.264/AVC decoder is able to extract the basis view and decode it correcly. In addiction to the NAL units defined in H.264/AVC, MVC defines a new kind of NAL unit, named coded slice of MVC extension, that carries the data pertaining MVC structure. When a H.264/AVC compliant encoder reads this NAL, it simply discards them, since it doesn't recognize the header. It's introduced the prefix NAL unit, already in the H.264/SVC extension (Scalable Video Coding), that informs the MVC decoder about temporal information not carried in base NAL. Backward compatibility is still mantained because H.264/AVC decoder doesn't recognize the prefix NAL.

MVC exploits Sequence Parameter Set (SPS) and Picture Parameter Set (PPS) structures to carry information about prediction dependency between views.

1.6.3.2 Stream extraction

MVC must provide synctactical structures to allow the extraction of sub-streams from the main MVC including all the views and all the frames. In fact, depending on the application or on some scalability issue, not all the views or frames need to be transmitted. Free-viewpoint or interactive TV needs only a subset of all the views (that varies dinamically), whereas temporal scalability (*i.e.*, frame dropping) is needed to adjust the video bitrate. The syntax elements provided by MVC are two, view_id, which identify the view, and temporal_id, that denotes the rate belonging to frame. For example, in Fig.5.1 I frames have the lowest temporal_id, whereas b frames are the highest, in fact they are the first frames to be dropped in case of temporal scalability.

Another syntax element is priority_id. It provides a lightweight procedure to extract the desiderated sub-stream from the main stream. At each prior-

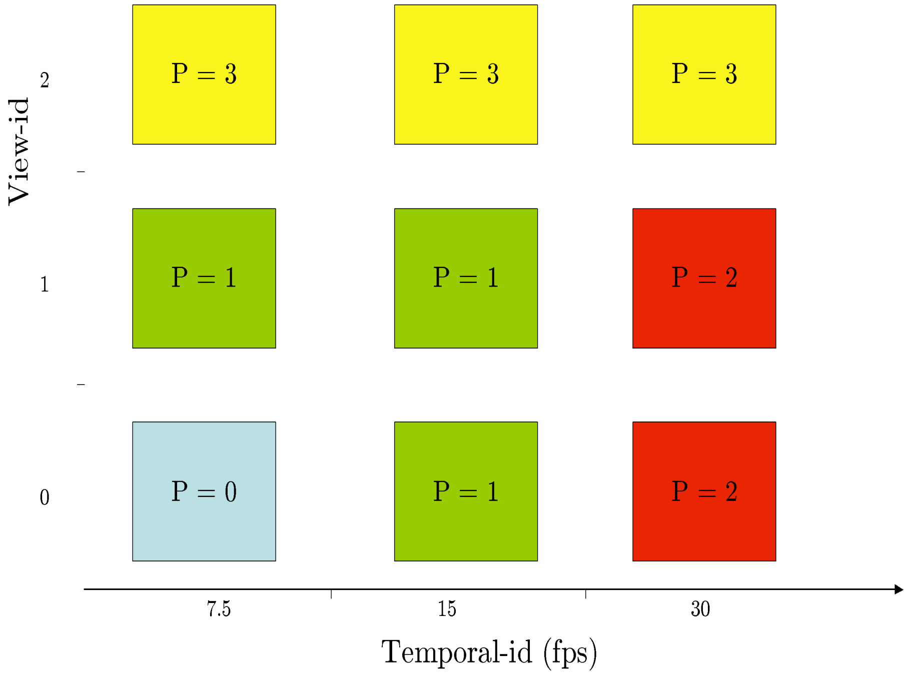

Figure 1.10: An example about how assigning a priority_id to the stream.

ity_id corresponds fixed values of temporal_id and view_id. To have a compliant MVC stream just extracting all the NALs having priority_id lower or equal to the desidered priority_id (*i.e.*, view_id and temporal_id). An example is given in Fig.1.10. The prority_id is indicated with the capital letter P: P = 0 is the base view at the lowest frame rate, P = 1 is 15 fps and view 0 and 1, P = 2 is 30 fps and view 0, 1 and finally P = 3 is the entire main MVC stream, *i.e.* 30 fps and all the views.

1.6.3.3 Decoding ordering of the stream

We call decoding order the ordered sequence of NAL units placed in the bitstream. With the respect to other video standards, MVC decoding order is more complicated and more strategies can be adopted because of the two dimensions, time and view. Principally, two kinds of ordering are considered: view-first and time-first ordering. In the view-first ordering, frames of the same view are transmitted sequentially within a GOP. Coversely, in the time-first ordering frames of different views but belonging to the same istant are transmitted sequentially within a GOP. Fig.1.11 and1.12 show respectively the view-first ordering and the time-first ordering.

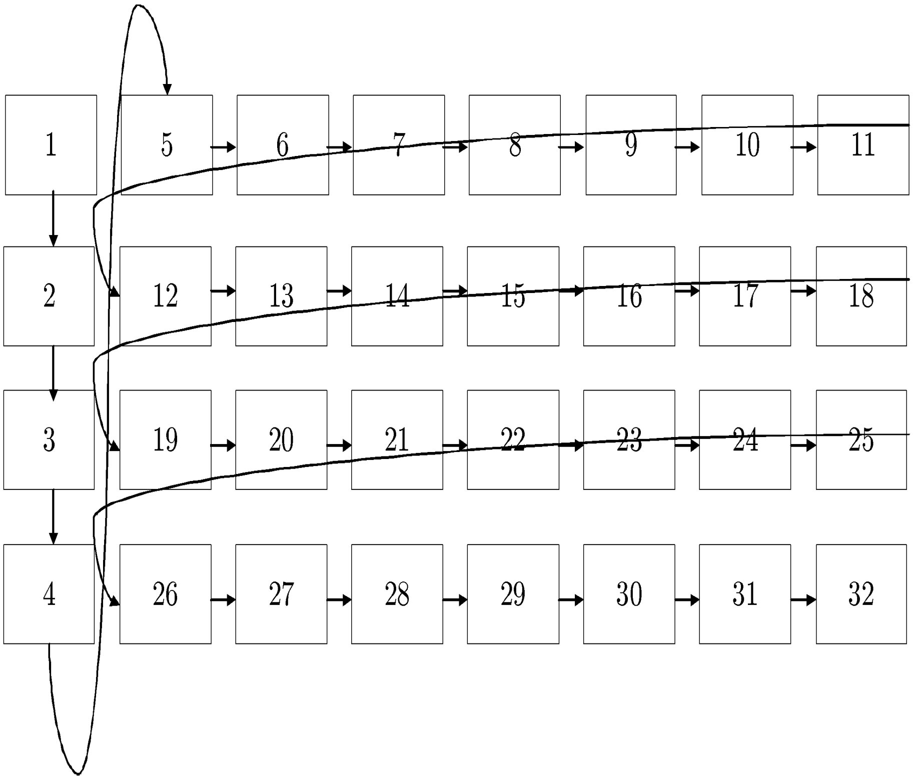

Figure 1.11: View-first ordering.

1.6.3.4 Parallel encoding

Because of the interview dependency, frames belonging to different views need to be decoded sequentially. This is a problem when all the views need to be decoded simultaneously, because the decoder must be N_{view} faster than a baseline decoder. To solve this is issue, *i.e.* parallelize the decoder procedure, in MVC is allowed signalling by a special message which areas are used for motocompensation. See Fig.5.1, view #0 is used as reference of view #1. The first row of view #1 is predicted only by the two first rows of view #0, whereas the second row of view #1 is predicted by the three first rows of view #0,

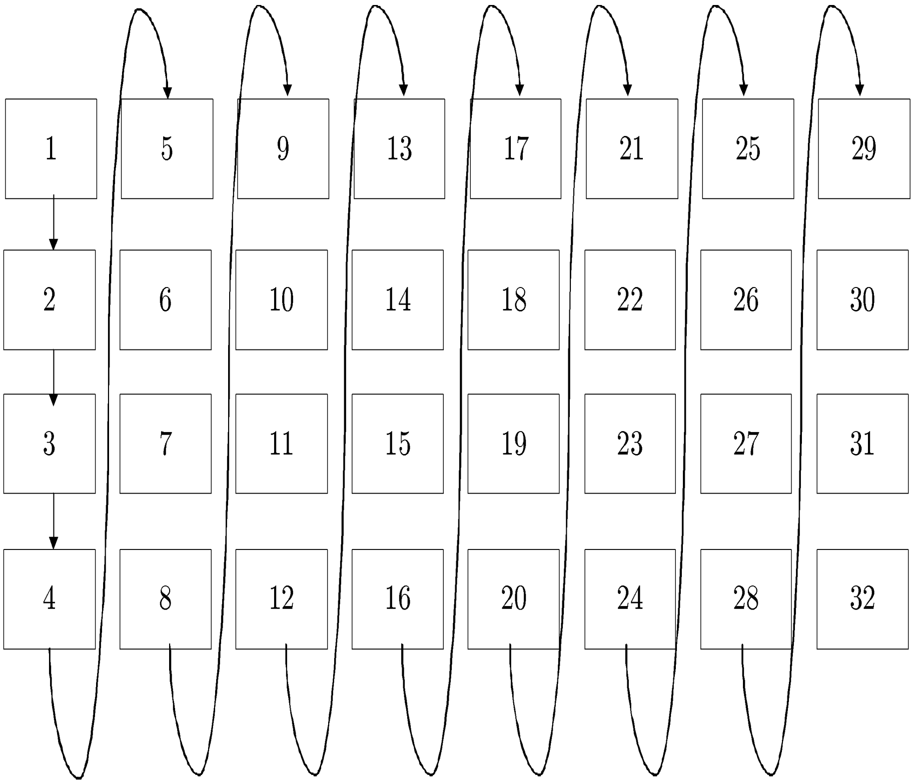

Figure 1.12: Time-first ordering.

1.7 Summary

In this chapter we have introduced the main topic of this thesis, mobile video streaming. A brief summary of the state-of-the-art of video encoder, H.264, and the introduction to the structural elements of a video streaming system are first offered. Then the introductions to the most relevant issues, that are the aims of this thesis are showed. In the next chapters we describe in more detail those issues and present our contribution.

Chapter 2

Improving Compression Efficiency: a Game Theoretic Approach

2.1 Introduction

Game theory is a branch of applied mathematics aimed at describing the behavior in strategic contexts, where an individual's success in making choices depends on the others' choices. Traditional applications of game theory try to find the equilibrium point of the game, that is a set of choices for each player such that none is likely to move from with an unilateral decision. A potential application of this optimization approach is found in source coding, and in particular in video coding, where a common bit budget is assigned and different visual data cooperate to maximize the overall quality of the video sequence. The former application of game theory in video coding is the work in (2), where the authors optimize the perceptual quality of the decoded sequence while guaranteeing "fairness" in bit allocation among macroblocks via a game theoretic approach. Since the whole frame is an entity perceived by viewers, macroblocks represent players that compete cooperatively under the global objective of achieving the best quality under the given bit constraint. This work has the merit to provide a first relation between visual quality and the utility function of the game theoretic approach. However, the work addresses a local spatial optimization of allocated coding resources, without considering the video sequence as a whole. In fact, game theoretic approach can be also employed to optimize higher level system constraints, such as random access points inter-distance and location or the video output buffer occupation and so on. Here, we resort to game theory to optimize not only the bit allocation between different frames of the video sequence, but also the choice of the optimal frame coding mode. The coding mode affects both the coding efficiency and the random access facilities of the coded video sequence. We entail the optimization in the framework of video streaming by means of the video coding standard ITU-T Rec. H.264, also known as ISO/IEC MPEG-4/Part 10-AVC.

State of the art works about video coding for bit-rate switching applications are based on realizing new coding schemes so to reduce the coding cost for SP frames. In (55) the authors propose a technique to improve the coding efficiency of the SP frames by limiting the mismatch between the references prediction and reconstruction. In (57) it is shown that by appropriately choosing reference pictures, the size of secondary SP frames can be reduced by up to 40% and 2% for random-access and rate-switching respectively, without affecting the PSNR. However, the analysis of the SP frames rate distortion curves and the comparison with analogous curves of Intra coded (I) and Predicted (P) frames show that the choice of the proper frame coding mode itself significantly affects the overall coding cost. As long as SP frames are concerned, due to the less favorable rate distortion characteristics, larger margins of bit saving - or, equivalently, of quality improvements - are expected by optimizing their allocation along the video sequence. In particular, given the maximum distance between SP frames, that can be considered as a system constraint depending on the desired degree of accessibility, there is still a degree of freedom on where to locate the SP frames among the sequence. Such a circumstance is here exploited to improve the quality of the overall encoded sequence by optimizing both the SP frames location and the bit budget allocated to each frame. Let us observe that, once the optimal frames budgets are allocated, the algorithm in (2) can be applied for individual frame coding. Experimental results show the performance of the optimized coding strategy in a video streaming environment allowing bitstream switching. In particular, the optimum resource allocation reduces the overall bitrate of the sequence still maintaining the same PSNR; moreover the number of bit per frame is much more equalized using the optimal coding algorithm, so resulting in a better utilization of the output buffer and in more equalized transmission delays.

The reminder of this chapter is organized as follows. Following the guidelines in (2), in Sect.2.2 we formulate the problem of video coding for bitstream switching by representing the frames of a sequence as players and the overall sequence quality as the objective function. The strategy of each player is the choice of the coding mode and the allocated bits. Sect.2.3 illustrates the algorithm for finding the optimum choice for the location of the SP frames, and for a fair distributions of bits per frame; the described approach can be performed offline, in order to provide pre-encoded sequence. Finally Sect.2.4 concludes the chapter.

2.2 Game theoretic approach to SP allocation

Game theory collects a set of analytical models with the goal of describing and characterizing situations (games) where more parties (players) dynamical interact to reach each one its own satisfaction (non cooperative game) or the collectivity satisfaction (cooperative games). The principles of this mathematical tool may apply in a great deal of fields as - for instance - social sciences (as economics), biology, engineering, etc. Basic assumptions over the behavior of each player are the following: *i)* a player acts so to maximize its own satisfaction (represented by a suitable utility function), *ii)* in selecting its strategy, a player accounts for the actions that the other players have chosen or are likely to choose.

In this contribution, we resort to a game theoretic optimization to address the problem of video coding for bit-stream switching applications. We investigate how the degree of freedom about the location of the SP frames in the encoded bitstream can be exploited to maximize the video quality of the encoded sequence under a given bit budget.

Here we propose a game theoretic approach to drive this maximization. In details, following the approach in (2), we recast the problem of frame type selection and bit allocation in terms of strategy selection in a cooperative game.

To cope with the system constraints on the desired degree of accessibility, the maximum distance N between switching frames is assumed to be given as a function of the maximum temporal distance $\tau_{\max}$ between SP frames and of the video sequence frame rate f_0, *i.e.* $N = f_0 \cdot \tau_{\max}$. Hence, the overall video sequence is divided in shorter sub-sequences of N frames. To satisfy the constraint on the maximum distance, in each sub-sequence at least one frame must be a switching one. This partition extends the classical structure of Group of Pictures (GOP) exploited in different video coding techniques to a more general and flexible structure.

Let us consider coding a target bit-rate of $R(bit/s)$ to encode the N frames window. The overall bit budget available for the N frames is $B = RN/f_0(bit)$.

The game is described as follows:

- the players of the game are the N frames of the sub-sequence;
- the strategy of each player is given by its coding type and by the number of bits allocated to itself;
- the utility of each player, representing its preference, is its visual quality after decoding.

Let us denote by $c_i, i = 0, \cdots N-1$ the frame coding type, where c_i takes value in a finite set C representing all the N_C coding modes provided by the video encoder, and r_i the number of bits allocated to itself. Due to coding constraints, the set $\mathcal{A}$ of admissible coding mode N-ples is included in C^N, and its cardinality $N_{\mathcal{A}}$, representing the overall number of different coding mode configurations, is smaller than N_C^N. For example, in the case of c_i representing a binary choice between P coding mode and SP coding mode, and assuming that only one out of N frames is a SP frame, in each window of N frames only N different SP frame coding locations need to be considered and $N_{\mathcal{A}} = N$.

Finally, let $u_i = u_i(c_i, r_i)$ denote the utility of the i-th player, *i.e.* the visual quality of the i-th frame, $i = 0, \cdots N-1$. Each player is characterized by the initial utility u_i^0, that represents the minimal visual quality that must be guaranteed, and by the corresponding number of allocated bits r_i^0 required to achieve the quality u_i^0. The tuples $< u_0, \ldots, u_{N-1}, u_0^0, \ldots, u_{N-1}^0 >$ represent the *game settings.*

The so-called Nash Bargaining Solution (NBS) is a unique solution that satisfies a set of axioms (efficiency, linearity, independency of irrelevant alternatives and symmetry) (45). In the video coding framework the NBS can be found by maximization of the following gain function (52):

$$\mathcal{G}(c_0, r_0, \cdots c_{N-1}, r_{N-1}) = \Pi_{i=0}^{N-1} \left(u_i(c_i, r_i) - u_i^0\right) \tag{2.1}$$

with respect to c_i, r_i under the constraints:

$$\begin{aligned} & r_i \geq r_i^0,\ i = 0, \cdots N-1 \\ & \sum_{n=0}^{N-1} r_i \leq B \\ & \{c_0, \cdots, c_{N-1}\} \in \mathcal{A} \end{aligned} \tag{2.2}$$

The visual quality $u_i = u_i(c_i, r_i)$ of the i-th frame is a value related to subjective perception, possibly affected by interaction between different media, and it is therefore hardly captured by an analytical relation (see (25; 51) for a comprehensive survey on the subject). The approach in (2), imposing a linear relation between the bit assigned to an image area, namely a macroblock, and its resulting visual quality after decoding, has the merit of leading to an analytically tractable solution for the maximization in (2.1). Hence, here we extend the relation formerly found in (2) by taking into account also the different coding efficiency corresponding to different frame coding modes and define the visual quality of the i-th frame as

$$u_i(c_i, r_i) = \frac{r_i}{K(c_i)\, g(\sigma_i)} \tag{2.3}$$

being σ_i the standard deviation of the innovation process between frame i and frame $i-1$, regarded as realizations of multidimensional random variables, and $g(\sigma_i)$ a non decreasing function[1]. The factor $K(c_i)$ represents the coding cost of the coding mode option associated to the i-th frame.

For any fixed set of coding modes $c_0, \cdots, c_{N-1}$, the values r_i optimizing (2.1) are proved to be (2):

$$r_i = K(c_i) g(\sigma_i) u_i^0 + \frac{1}{N}\left(B - \sum_{n=0}^{N-1} K(c_n)\, g(\sigma_n) u_n^0\right) \tag{2.4}$$

[1] (2) $g(\sigma_i) \overset{\mathrm{def}}{=} \sigma_i^{\alpha}, \alpha = 0.8$.

In assigning the minimal quality that must be guaranteed to each frame $u_i^0, i = 0, \cdots N-1$, different priors can be adopted. Let us suppose that, in order to avoid annoying fluctuations of the frame visual quality, uniform minimal quality all over the sequence is adopted, *i.e.* $u_i^0 = u_{\min}, i = 0, \cdots N-1$. Then, let us denote by $i_0, \cdots, i_{N-1}$ the indexes of the N frames ordered by increasing standard deviation of the innovation process, *i.e.*

$$\sigma_{i_0} \leq \cdots \leq \sigma_{i_{N-1}}$$

Let us consider the set of N_c admissible N-ples $c_i, i = 0, \cdots N-1$, and the corresponding N-ples of coding costs $K(c_i), i = 0, \cdots N-1$. It is easily shown that the gain in (2.1) is maximized by any admissible N-ple such that the coding modes $\overline{c}_i, i = 0, \cdots N-1$ satisfy the following ordered inequalities *i.e.*

$$\{\overline{c}_i, i = 0, \cdots N-1\} \text{ such that } K(\overline{c}_{i_0}) \geq \cdots \geq K(\overline{c}_{i_{N-1}}) \tag{2.5}$$

This solution correspond to the intuitive choice of assigning the less efficient coding modes to the frames with smaller standard deviations of the innovation process. Finally, should different minimal qualities be required for the N frames, the same criterion herein exposed applies to the indexes of the N frames ordered by increasing weighed standard deviation of the innovation process, *i.e.*

$$w_{i_0}\sigma_{i_0} \leq \cdots \leq w_{i_{N-1}}\sigma_{i_{N-1}}$$

being $w_i = u_i^0 / \min(u_{i_0}^0, \cdots u_{i_{N-1}}^0)$. Since this maximization is found whatever the effectively assigned bit budget is, the assignment of the coding mode can be made independently on the evaluation of the allocated rates r_i.

2.3 Optimal coding algorithm

In this section we present the coding algorithm optimized accoding to the criteria exposed in Sect.2.2 by discussing also a few implementation details.

2.3.1 Sequence partitioning

The coding optimization algorithm is applied by first partitioning the overall sequence in different shorter sequences of equal length N. In each sequence at least one SP frame shall be introduced. Since SP frames have less favourable rate-distortion than P frames, unnecessary SP frames should not be introduced. Hence, without loss of generality, we will assume that only one frame out of N is an SP frame and its locations is chosen by performing the optimization on each sequence, in different steps.

2.3.2 Innovation Process Variance Estimation

According to (2.5), the standard deviation σ_i of the innovation process of the i-th frame is estimated as the standard deviation of the motion-compensation residuals. We observe that the motion compensation residuals are generated during the coding process and vary depending on the compression ratio. However, we disregard the variations on the different range of compression ratios and we estimate the variance on the original sequence. This choice is also motivated by the fact that in realistic streaming system the primary SP frames of a sequence should be set in all the coded version of the same sequence at the same temporal index. Hence, in the following we will evaluate the standard deviation of the motion-compensation residual on the frames of the original sequence.

2.3.3 Coding Mode Assignment

Once the standard deviation has been evaluated, according to (2.5) the SP coding mode is assigned to the frame with the minimum standard deviation of the innovation process. The value of the parameter $K(c_i)$ can be estimated either by rate distortion curves at a typical distortion value, or assigned through a priori criterion.

2.3.4 Rate evaluation

After the choice of the coding mode of each frame, the preliminary matter assignment of the initial rates r_i^0 is performed, based on the initial assignment of the qualities $u_i^0, i = 0, \cdots N-1$. Recent investigations on the theoretical and experimental rate-distortion performance of SP and P frames have highlighted that a given level of distortion is achieved by higher rate for SP frames than for P frames (50). Hence, to equalize the initial quality, a large initial bit budget r_i^0

is assigned to the SP frame; the $r_i, i = 0, \cdots N-1$ are then straightforwardly evaluated using (2.4).

2.3.4.1 Frame Coding

Given the frame bit budget r_i obtained at the previous step, individual frame coding can be performed resorting to state of the art optimization algorithms as, for instance, the one presented in (2). It's also possible to assign a single quantization parameter value to the whole frame. In this case the quantization parameter is chosen as the minimum integer value compatible with the assigned value r_i.

2.4 Summary

In this chapter we have illustrated a way to efficiently encode a video stream with SP frames by means of game theory. We relaxed the usual assumption of a fixed GOP by varying the position of the SP frame in the GOP. Using Nash Bargaining Solution the SP is allocated in the frame exhibiting the minimum innovation, and the bit budget is allocated according to a "fair" distribution. We lack of experimental results because we don't have and heuristic procedure to choose the correct QP per frame/macroblock in order to assign the bit budget to every frame. In this sense, we could apply existing rate control techniques.

Chapter 3

Improving Error Resilience: a Multiple Description Coding Approach

3.1 Introduction

Multiple description coding (MD Coding, or MDC) consists in providing different coded description of the same data, and to send them using different transport channels, achieving the transmission diversity needed for error resiliance. Surveys on principles in designing MD video coders and on different MD compression algorithms can be found in (23; 61). MDC algorithms descriptions are generated by subsampling the data either in the spatial, temporal, or frequency domain. A possibly lost description can be estimated from the others by exploiting spatial or temporal adjacent video data samples correlation. In (21), two MDC algorithm based on polyphase down-sampling are investigated and their performances over unreliable networks assessed by numerical simulations. In (59) the authors analyze a mathematical framework for pre- and post-processing two descriptions of the original data, so as to implement the MDC paradigm by exploiting the native directional correlation characteristics of the image. Specifically, in the pre-processing stage the data splits into two subsets by means of a forward transform, that are separately encoded and transmitted. At the receiver side, data is recovered by an inverse transform making use only of the effectively available description. In (5; 60), MDC is achieved by originating four descriptions from the spatially downsampled polyphase components of the original frames; each description is independently H.264 coded, and concealment is applied at the decoder side in case of losses. In (32) a distributed video streaming framework using unbalanced MDC and unequal error protection for wavelet-based coders is proposed. In (54), a MDC technique based on the H.264/AVC slice group syntactic structure (63) is described. Recently, in (58) a novel MDC technique has been proposed, in the framework of H.264 coding. The coding algorithm exploits the H.264 redundant slices; at the receiver side, the received compressed bitstream must be pre-processed before being applied at the input of a standard compliant decoder.

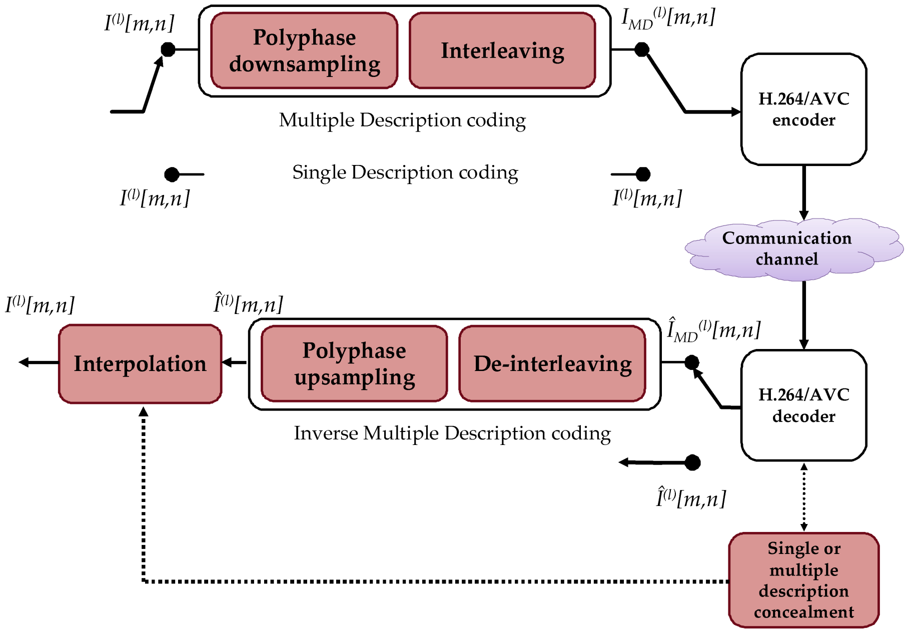

Figure 3.1: MDC scheme proposed.

In this chapter we analyze an error resilient MDC scheme based on Polyphase SubSampling (PSS MDC). The analyzed MDC scheme is shown in Fig.3.1. The original video sequence is applied at the input of a spatial-temporal interleaving stage that generates a synthetic sequence, in which each frame conveys a fixed number of descriptions pertaining to different frames of the original sequence. The synthetic sequence is applied at the input of a standard video encoder; the transport channel diversity is achieved simply by mapping different encoded frames into different transport packets. Let us observe that, since MDs are managed only inside the interleaving stage, switching from a MDC scheme to a Single Description (SD) coding one can be performed dynamically at the encoder input at the expence of an interleaving delay, easily recoverable by suitable buffering at the decoder side. At the receiver side, the synthetic sequence is decoded and concealed using available MD and de-interleaving is applied to provide a coarse reconstruction of the original video sequence.

Once this coarse estimate of the video sequence has been provided, a fast restoration algorithm based on robust interpolation is applied. The interpolation algorithm exploits the local image directionality feature as well as the information on which descriptions have been correctly received and which have been concealed and it effectively improves the decoded video sequence quality both from an objective and from a subjective point of view. The herein analyzed scheme presents two interesting properties: the MDC technique employs a standard compliant video coding stage, so that it can be implemented at a slightly increased computational cost; besides, it appears to the transport layer as a SD coded data flow, and the increase of protocol overhead is limited, too.

The remainder of the chapter is organized as follows: in Sect.3.2 the overall MDC coding scheme is outlined, while in Sect.3.3 the standard compliant encoding/decoding stages providing a coarse video sequence reconstruction are described. Sect.3.4 describes the final restoration stage performing a robust edge-preserving interpolation; the results of numerical simulations assessing the MDC algorithm performance are shown in Sect.3.5. The paper is concluded in Sect.3.6.

3.2 MD Generation

PSS MDC generates descriptions of each single frame by subsampling it with different initial phases in vertical and/or horizontal directions; each subsampled image provides a simplified frame description, and the original frame is recovered by suitably collecting different descriptions.

Benefits of PSS MDC are achieved when the different descriptions are transmitted in diversity. However, independent MD encoding and transmission not only results into heavier computational requirements due to encoding and decoding different descriptions, but also into increased protocol overhead and reduced bandwidth efficiency[1] Here, we describe how PSS MDC can be realized by an application layer interleaving scheme operating at the input of a standard compliant encoding module, so as to bound the computational requirement. Furthermore, the interleaving generates a synthetic sequence in which each frame contains different subsampled descriptions pertaining to different frames of the original video sequence; thus, diversity is straightforwardly achieved when each coded frame of the interleaved sequence is mapped into at least one transport-layer packet.

Let us denote the l-th frame of the original video sequence, of dimensions $M \times N$ by

$$I^{(l)}[m,n],\ m = 0, \cdots, M-1, n = 0, \cdots N-1$$

and let us suppose that a $K \times K$ downsampling factor is employed in the PSS stage, so that each frame is conveyed by $K \times K$ descriptions. The j-th poliphase subsampled description of the l-th frame, of size $M/K \times N/K$, is given by

$$\begin{aligned} &\Delta_j^{(l)}[m,n] \overset{\text{def}}{=} I^{(l)}[Km + m_j, Kn + n_j], j = 0, \cdots K^2 - 1 \\ &m_j = j_{\text{mod}K}, n_j = \lfloor j/K \rfloor, \\ &m = 0, \cdots, M/K - 1, n = 0, \cdots N/K - 1. \end{aligned} \tag{3.1}$$

The K^2 descriptions can be juxtaposed into a single $M \times N$ frame, so as to associate at the original video sequence a new spatially interleaved sequence:

$$\begin{aligned} &I_{\text{SI}}^{(l)}[m + m_j * M/K, n + n_j * N/K] \overset{\text{def}}{=} \Delta_j^{(l)}[m,n], \\ &m = 0, \cdots, M/K - 1, n = 0, \cdots N/K - 1, j = 0, \cdots K^2 - 1, \end{aligned} \tag{3.2}$$

The spatially interleaved sequence $I_{\text{SI}}^{(l)}[m,n]$ exhibits more rapid luminance variations than the original sequence $I^{(l)}[m,n]$, thus presenting a higher coding cost. However, since in video coding intensive prediction techniques are used, the overall coding cost is strongly related to interframe correlation, and a suitable temporal interleaving increases the coding efficiency. Then, on the spatially interleaved sequence, a temporal interleaving is applied, aiming at

[1] In almost all the emerging video communication schemes, it is recommended a one-to-one correspondence between application layer packets and transport packets (48), resulting in protocol overhead in case of independent MD transmission.

- assigning different descriptions of the j-th frame to different application layer packets
- preserving the inter-frame correlation properties typical of natural video sequences.

The spatio-temporal interleaved sequence is built as

$$I_{\mathrm{MD}}^{(l)}[m + m_j * M/K, n + n_j * N/K] = \Delta_j^{(l+j)}[m, n] \tag{3.3}$$

for $m = 0, \cdots, M/K - 1, n = 0, \cdots N/K - 1, j = 0, \cdots K^2 - 1$ From (3.3), we recognize that the l-th frame of the interleaved sequence $I_{\mathrm{MD}}^{(l)}[m, n]$ conveys K^2 subsampled descriptions, with different sampling phases, pertaining to K^2 different frames of the original sequence $I^{(l)}[m, n]$, namely $l, l + 1, \cdots, l + K^2 - 1$ of the original sequence $I^{(l)}[m, n]$; conversely, the K^2 descriptions of each frame of $I^{(l)}[m, n]$ are conveyed by K^2 different frames of $I_{\mathrm{MD}}^{(l)}[m, n]$. Thus, the interleaving introduces diversity when each frame of the interleaved sequence $I_{\mathrm{MD}}^{(l)}[m, n]$ is sent using a different transport packet; this condition is a minimal requirement (48) that is expected to be satisfied by all video communication systems. Such interleaving also preserves the inter-frame correlation, in fact each description follows the correspondant description of the previous frame.

For instance, let us fix $K = 2$. Then, given the original sequence $I^{(l)}[m, n]$, the PSS followed by the interleaving stage generates the MD sequence $I_{\mathrm{MD}}^{(l)}[m, n]$:

$$\begin{array}{c}
\begin{array}{l|cc||cc||cc||cc|l}
\cdots & \Delta_0^{(l-1)} & \Delta_1^{(l)} & \Delta_0^{(l)} & \Delta_1^{(l+1)} & \Delta_0^{(l+1)} & \Delta_1^{(l+2)} & \Delta_0^{(l+2)} & \Delta_1^{(l+3)} & \cdots \\
\cdots & \Delta_2^{(l+1)} & \Delta_3^{(l+2)} & \Delta_2^{(l+2)} & \Delta_3^{(l+3)} & \Delta_2^{(l+3)} & \Delta_3^{(l+4)} & \Delta_2^{(l+4)} & \Delta_3^{(l+5)} & \cdots
\end{array} \\
\\
\begin{array}{ccccc}
 & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\
\ldots & \text{frame } l-1, & \text{frame } l, & \text{frame } l+1, & \text{frame } \ l+2, \quad \ldots
\end{array}
\end{array}$$

whose l-th frame is built by juxtapposing the $K^2 = 4$ descriptions $\Delta_0^{(l)}, \Delta_1^{(l+1)}, \Delta_2^{(l+2)}, \Delta_3^{(l+3)}$ pertaining to the frames $l, l + 1, l + 2, l + 3$ of the original video sequence.

3.3 MD encoding and decoding

The MD sequence $I_{\mathrm{MD}}^{(l)}[m, n]$ is applied at the input of a standard video encoder, and transmission diversity of MD is implicitly assured when each coded frame is mapped into at least one independent transport packet. From now on, and without loss of generality, we will refer to a video communication scheme based on the most recent Joint Video Team coding standard H.264 (65). In the H.264 framework, each frame can be coded in one or more Network Adaptation Layer Unit (NALU), and each NALU can be coded in one or more transport packets. In either case, coded video data pertaining to different frames are expected to be conveyed by different transport packets when IETF recommended packetization (62) is applied. Hence, from a technical point of view, the herein analyzed MDC scheme has the merit to bound the visibility of MDC to the application layer and to appear to the network as a unique media flow, so simplifying the protocol architecture required for the video communication.

At the receiver side, the video stream is decoded; in case of channel errors, data are lost and the decoder invokes error concealment procedures to provide the decoded video sequence $\hat{I}^{(l)}_{\mathrm{MD}}[m,n]$. In case of data loss, error concealment may use the same algorithms adopted in SD coding or exploit the MDC paradigm by searching in the decoder buffer for alternative descriptions of lost data. When the loss occurs on less than K^2 consecutive frames, the decoder can exploit the availability of MD pertaining to the same frame of $I^{(l)}[m,n]$ to recover the loss; otherwise it performs a generic concealment, for instance by exploiting descriptions belonging to adjacent frames.

Once the sequence $\hat{I}^{(l)}_{\mathrm{MD}}[m,n]$ has been generated, de-interleaving is applied to generate the decoded version $\hat{I}^{(l)}[m,n]$, which represents a coarse estimate of the trasmitted sequence $I^{(l)}[m,n]$.

3.4 Video sequence restoration by means of robust edge preserving interpolation

After the decoding, error concealment and de-interleaving stages, the reconstructed video sequence $\hat{I}^{(l)}[m,n]$ is available. The luminance values $\hat{I}^{(l)}[m,n]$ may be error-free or affected by reconstruction errors, resulting from losses of coded data pertaining to the l-th frame or from propagation of errors occurred on preceding frames due to the employment of predictive coding. Moreover, the amount of error varies from a pixel to another. In fact, at the output of the decoding and de-interleaving stage, in $\hat{I}^{(l)}[m,n]$ we can distinguish

- error-free pixels,
- pixels that have been concealed using at least one correctly received description pertaining to the same frame
- pixels that have been concealed in absence of alternative descriptions pertaining to the same frame.

Hence, we recognize that the MDC concealment induces a fairly natural reliability hierarchy in the luminance values of the pixels of the sequence $\hat{I}^{(l)}[m,n]$. We formalize this hierarchy by introducing three classes of pixels, namely Class I (error-free), Class II (MD concealed), and Class III (SD concealed) and assigning a different reliability $r^{(l)}[m,n]$ to pixels belonging to different classes. Then, the decoding sequence quality can be improved by applying a restoration algorithm that takes into account not only the local image edges but also the pixel reliability. The restoration stage operates by replacing each concealed (Class II or Class III) pixel in $\hat{I}^{(l)}[m,n]$ with a suitably interpolated estimate.

The herein presented interpolation technique extends the classical edge-detection interpolation scheme Edge-based Line Average (ELA) into a Robust ELA (RELA) to exploits the reliability of the descriptions available for interpolation. Namely, for a given site (m,n), we define a 3×3 neighborhood $\eta(m,n)$ as illustrated in Fig.3.2. Then, for each site (m,n), four pairs of pixels belonging to $\eta(m,n)$ are individuated as: $\{(m+\delta_v, n+\delta_h),(m-\delta_v, n-\delta_h), \delta_v, \delta_h \in \mathcal{S}_{\mathrm{ELA}}\}$ being $\mathcal{S}_{\mathrm{ELA}}$ defined as: $\mathcal{S}_{\mathrm{ELA}} \overset{\mathrm{def}}{=} \{(1,0),(0,1),(1,1),(1,-1)\}$. Each pair of pixels, indexed by δ_v, δ_h, identifies a candidate direction for interpolation; ELA searches for the direction of minimal luminance variation, *i.e.* for the pair (δ_v, δ_h) minimizing $\left|\hat{I}^{(l)}[m+\delta_v, n+\delta_h]-\hat{I}^{(l)}[m-\delta_v, n-\delta_h]\right|$ and estimates the luminance in $\hat{I}^{(l)}_{\mathrm{ELA}}[m,n]$ as the average between $\hat{I}^{(l)}[m+\delta_v, n+\delta_h]$ and $\hat{I}^{(l)}[m-\delta_v, n-\delta_h]$. Since it has been originally designed for fast upsampling of high quality images, the ELA interpolation algorithm does not take into account possible errors affecting the luminance of the pixels in $\eta(m,n)$; thus, it performs quite well on reconstructing missing pixels for error free descriptions, but it presents modest performances when the descriptions are affected by residual errors after concealment.

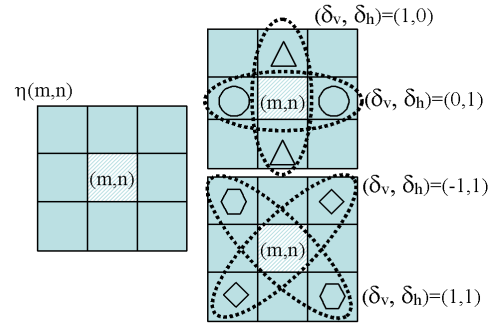

Figure 3.2: The considered neighborhood $\eta(m,n)$ and the associated four pixel pairs, indexed by $(\delta_v, \delta_h) \in \mathcal{S}_{\mathrm{ELA}}$.

$$\mathcal{S}_{\mathrm{RELA}} = \left\{(\delta_v,\delta_h) \in \mathcal{S}_{\mathrm{ELA}}, (\delta_v,\delta_h) \text{ s.t. } \left|r^{(l)}[m+\delta_v, n+\delta_h]\right| + \left|r^{(l)}[m-\delta_v, n-\delta_h]\right| > \theta\right\} \tag{3.4}$$

$$(\delta_v^{(\mathrm{RELA})}, \delta_h^{(\mathrm{RELA})}) = \underset{(\delta_v,\delta_h)\in\mathcal{S}_{\mathrm{RELA}}}{\arg\min} \left|\hat{I}^{(l)}[m+\delta_v, n+\delta_h] - \hat{I}^{(l)}[m-\delta_v, n-\delta_h]\right| \tag{3.5}$$

$$\hat{I}^{(l)}_{\mathrm{RELA}}[m,n] = \frac{\hat{I}^{(l)}[m+\delta_v^{(\mathrm{RELA})}, n+\delta_h^{(\mathrm{RELA})}] + \hat{I}^{(l)}[m-\delta_v^{(\mathrm{RELA})}, n-\delta_h^{(\mathrm{RELA})}]}{2} \tag{3.6}$$

We design here a robust edge driven interpolation algorithm (RELA) taking into account the measurements reliability. Specifically, let us assume that a reliability measure $r^{(l)}[m,n]$ is associated to each pixel $[m,n]$ of the l-th frame after the concealment stage. This measure is used to operate a reduction of the set of directions which are candidate for interpolation, by limiting to the set of most reliable direction, *i.e.* the set $\mathcal{S}_{\mathrm{RELA}}$ as in (3.4), being θ a suitably defined threshold. Then, the optimal interpolation direction is determined as in (3.5), and estimates the luminance in (m,n) as in (3.6). From (3.6), we recognize that the robust interpolation attempts to restore the concealed pixels by directional smoothing, meanwhile using only the most reliable luminance values. Finally, we observe that the described interpolation strategy estimates the luminance value at the location (m,n) employing directional interpolation of pixels belonging to $\eta(m,n)$. The reformulation of this interpolation in terms of Bayesian interpolation of Markov Random Fields is currently under investigation.

3.5 Numerical Results

In this section we present a set of numerical simulation results assessing the performance of PSS MDC technique using the robust RELA interpolation. The experiments refer to the test sequences *Foreman* and *News*, CIF format, at 10 frames per second.. A number of $K = 2 \times 2$ descriptions has been selected. The RELA was realized by assigning $r^{(l)}[m, n] = 2$ to Class I pixels, $r^{(l)}[m, n] = 1$ to Class II pixels, $r^{(l)}[m, n] = 0$ to Class III pixels, and by setting $\theta = 1$.

The interleaved sequences are encoded using the reference JM H.264 coder version 11.0 (24), one NALU per frame. The GOP structure is given by a primary SP frame followed by 9 P frames; in each frame, 40 macroblocks are INTRA encoded for Random Intra Refresh purposes. The H.264 Video Coding Layer encodes one slice per frame, and the Network Adaptation Layer followed by the RTP packetizer using the so called simple packetization method maps each slice into an RTP packet.

The first set of numerical simulations refers to the encoding of 100 frames of the sequence News, at a bit-rate of 600 kbps. We analyze here in detail a run characterized by PLP=13%. MDC using RELA reduces the visually relevant artifacts that are observed on the decoded video sequence in presence of transmission errors. Fig.3.3 shows selected details of a few snapshots captured within the sequence decoded using ELA and RELA; the visual quality improvement achieved by adopting MDC with RELA is clearly appreciated. The quality of the video sequences decoded in different conditions has been also evaluated in terms of Peak to Signal Noise Ratio (PSNR), defined as: $\mathrm{PSNR} \overset{\mathrm{def}}{=} 255^2/\mathrm{MSE}$, proving that the RELA stage significantly improves the overall MDC performance, resulting into a PSNR gain of 1.5 dB over MDC without interpolation and 1.1 dB over ELA.

The second set of numerical simulations refer to the encoding of 100 frames of the sequence Foreman, at a bit-rate of 750 kbps. A transport channel characterized by a packet loss probability (PLP) equal to 10% has been simulated over 100 Montecarlo runs. The MDC scheme using RELA, ELA and MDC without interpolation have been compared, by evaluating the decoded sequence PSNR values observed on each of the 100 frames, and by characterizing statistically the PSNR values observed using the different schemes. Fig.3.4 reports the PSNR histograms of the three schemes, while Table 3.1 reports selected parameters characterizing

PSNR	MDC using RELA	MDC using ELA	MDC w/out interpolation
Mean	30.59	29.41	28.14
Standard Deviation	3.42	4.03	4.56
Median	31.03	29.41	27.63

Table 3.1: PSNR of the MD encoded sequences, decoded using R-ELA and ELA interpolation, and without interpolation: Foreman sequence, CIF, 750 kbps, 100 encoded frames, 50 Montecarlo runs.

the PSNR distributions.

3.6 Summary

In this chapter, we have analyzed a PSS MDC scheme. The underlying assumption of general MDC schemes is that each description is a tight approximation of the others; in the particular case of PSS MDC, this is the consequence of the correlation between the neighboring pixels of a natural image. In case of losses, the availability of multiple descriptions is exploited to perform a more accurate error concealment. The concealment induces a fairly natural pixel hierarchy that can be exploited by a post-processing stage. Here, we analyze a fast restoration stage that makes use not only of the local directionality information but also of the reliability of the decoded pixes. The scheme effectively improves the decoded video sequence quality on lossy channels both from an objective and from a subjective point of view. The interesting performance of the interpolation stage are related to the markovian nature of natural images; the relation between the interpolation algorithm and markovian image interpolation (41) is currently under investigation. The herein presented MDC scheme is realized in the form of pre-processing and post-processing stage in H.264/AVC standard compliant encoder and decoder pair; besides, transmission diversity is straightforwardly obtained by mapping one application packet in at least one transport packet, while maintaining a single transport data flow.

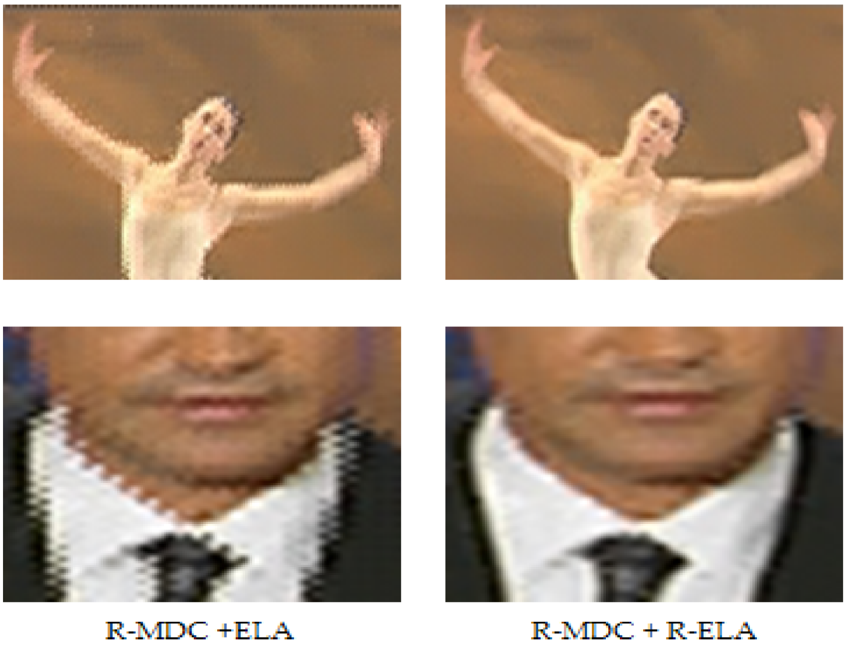

Figure 3.3: Details for frame 46 of the sequences decoded using R-ELA and ELA interpolation, News sequence, CIF, 600 kbps.

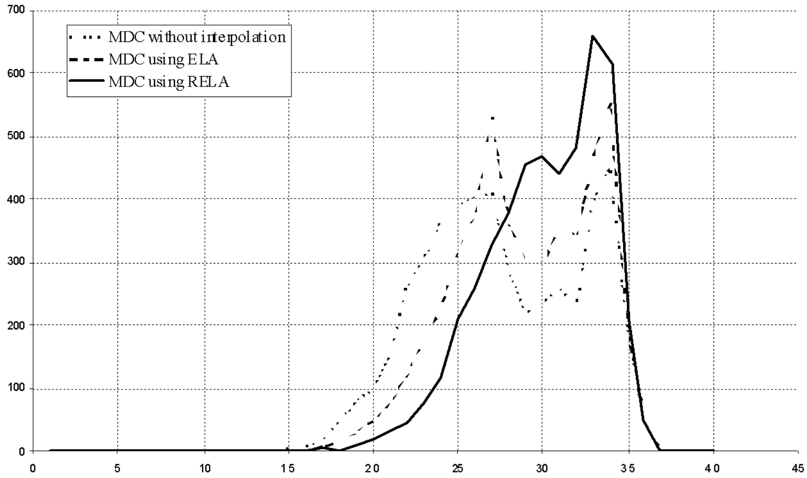

Figure 3.4: PSNR Histograms for the MD encoded sequences, decoded using R-ELA and ELA interpolation, and without interpolation: Foreman sequence, CIF, 750 kbps, 100 encoded frames, 50 Montecarlo runs.

.

Chapter 4

Improving network dimensioning: a Markovian source model

4.1 Introduction

In this work, we address the modeling of a H.264 video streaming source that performs bitstream switching using SP frames. Modeling of video data is useful in many respects. For instance, since video is typically the most expensive media in bandwidth allocation, proper simulations of the network load may require a reasonably high number of video data traces, and such a huge measurement in a variety of encoding conditions is a heavy task, especially when bandwidth adaptation is required. Conversely, having a compact video source model permits to numerically generate a large amount of realistic data from only few parameters or to elaborate those statistics of the video traffic having a major impact on the network. Besides, a theoretical model of the video source offers a clear analytical framework for the design of call-admission-control procedures as well as of cross-layer optimization strategies. Additionally, the model can be exploited to optimize the resources' allocation, especially in case of dynamic networks such as wireless mesh networks or cognitive radio networks.

Video source modeling has been widely investigated, with particular reference to videoconferencing services, which employ strictly constrained Constant Bit Rate (CBR) sources, or multicasting/broadcasting digital television oriented services, which envolve loosely constrained or unconstrained Variable Bit Rate

(VBR) sources. A survey on the subject is found in (28) where an overview of different video source models is presented, and the so-called Markovian Transform Expand Sample (TES) as well as self-similar models are investigated. Frame level statistical descriptions are found in (27), (47), (68) where the authors use a Markov chain to model the number of Asynchronous Transfer Mode (ATM) cells per frame in videoconference service, being each chain state representative of the number of cells. In (47), the frame size is modeled as the sum of three processes, two of which are first order autoregressive (AR) processes modeling the autocorrelation function at short lags, while the third process is the output of a Markov chain modeling the scene changes; specifically, three states are present, respectively modeling the first frame where scene change happens, the following frame, and all the other frames. In (68) a switching Markov AR process models the frame sizes, but at each state change the memory of the AR process is neglected; the model adopts a mixture of Gaussian densities for the pdf of the frame size. In (42), the video source model consists in nested AR processes modeling the mean and the fluctuations of the frame sizes in different scenes. Scene length is modeled by a geometric distribution. In (3) a video sequence is decomposed according to the motion/scene complexity and each part is described by a self-similar process. Beta distribution is used to characterize the marginal cumulative distribution of the self-similar processes. In (16), addressing cable digital television services, an MPEG1 video source is synthesized by a compound model involving three different discrete time AR processes, one for each kind of frame (I, P and B) employed in the standard, and a Markov chain representing the video activity. Models of H.264 video source have been considered in more recent papers, such as (33) and (34), in which the authors analyze a Markovian representation of an H.264 stream based on a Gamma like marginal frame size distribution to fit the I, P, and B distribution. In (37), the authors investigate about the marginal distribution of the frames and the scene changes, modeling the scene duration as a geometric distribution while in (17) a representation of an MPEG4 and H.264 VBR source is developed via a wavelet and time domain combined method that addresses the correlation of the encoded data referring to groups of consecutive frames beginning with a random access unit.

However, the aforementioned literature lacks of analysis of the dynamical behavior of the H.264 source performing bitstream switching; a few contributions on this issue can be found in (12)-(14). In (12), a Markov chain models the whole frame sequence by representing a frame as a state in the chain. In (13), the authors describe encoded data representing a group of consecutive frames beginning with a random access unit (Group of Pictures, GOP) as the output of a switching autoregressive hidden Markov process whose states represent different kinds of GOP. Instead, a low order autoregressive process is used in (14) to model the correlation between the frames, whereas a Markov chain drives the global averages at the GOP layer. Moreover, while the literature offers several Markovian models, the estimation of the model parameter set is often conducted in accordance to specific criteria (least squares estimation, method of moments, etc.) designed on a heuristic basis for the problem under consideration. A Markovian model enabling parameter estimation via local maximization of the likelihood function appears in (15). Herein, the preliminary results presented in (15) are extended and the model is investigated more in depth.

This chapter aims at modeling the video streaming traffic generated by a H.264 video encoder, which dynamically varies its rate by means of bitstream switching; furthermore, we are interested in designing an optimal strategy for parameter estimation from a real video sequence in accordance to a Maximum Likelihood (ML) estimation criterion. The herein analyzed model describes a random vector representing the sizes of the frames in a GOP. Since the bitstrem switches happens on network/client feedbacks depending on the channel status, such model includes implicitly a partial channel model. The basic idea is to provide a general-purpose model, that is able to describe efficiently the video source on different channels. At this aim, we assess the model employing an EDGE channel model (**?**). Let us remark we do not consider the EDGE channel model during the development of the model. We model the video source by resorting to a Hidden Markov Process (HMP) that describes the data vectors representing GOP frame sizes as the output of an underlying Markov chain. Each state of the chain is representative of a different kind of GOP, and the state-dependent conditional observation probability density function (pdf) is characterized by modulating the mean and the covariance matrix of a multivariate white Gaussian process according to the state parameters. We will show that this HMP turns out to be a Gaussian Mixture Process with Markov dependence, for which stationarity and ergodicity are ensured under mild assumptions on the underlying Markov chain. First and second order statistics of the HMP have been evaluated. Furthermore, we have outlined the ML parameter estimation by application of the Expectation-Maximization (EM) algorithm (18) to the case under study; thus, we have devised a procedure for estimation of the HMP parameters from a finite set of measurements extracted from a real video source, and in absence of *a priori* knowledge on the switching source behavior. Finally, we have assessed the model performance by comparing the traffic generated by an H.264 video source (in an EDGE system) with the synthetic traffic generated by a HMP using parameters estimated by means of the EM algorithm on a subset of the video data. We have also carried out comparisons of different statistics measured on the real video data with their expected value analytically derived in accordance to the HMP models. Furthermore, we have compared the buffer load, in terms of frame loss rate, for the real and the synthetic source. The results show that, despite of a few simplifying modeling assumptions, the HMP model well captures the statistical characteristics of the source at a GOP Layer; meanwhile, the application of the EM algorithm to the case under study provides a theoretically clear and technically sound framework for the HMP parameters tuning stage.

The remainder of this chapter is organized as follows: in Sect.4.2 we model the video source resorting to Hidden Markov Processes (HMPs) and in Sect.4.3 we devise a parameter set estimation procedure based on the EM algorithm. In Sect.4.4, we present numerical simulations assessing the convergence of the EM algorithm to the true HMP model parameters in a realistic scenario and we show that the HMP model using estimated parameters achieves satisfying performance in capturing the significant statistical characteristics of the data encoded by a real H.264 source. Conclusion is drawn in Sect.4.6.

4.2 The Hidden Markov Process

HMPs are a family of stochastic processes describing a discrete-time finite-state homogeneous Markov chain observed through a discrete-time memoryless invariant channel, where the adjective “hidden” denotes that the state sequence is never observed directly. These models have been deeply studied, and a review of a variety of HMP can be found in (20), where different stationarity and ergodicity conditions and algorithms for parameter estimation are also discussed.

Let us denote by $x[m]$ the size in bit of the m-th frame of the coded video sequence, and by $\mathbf{x}[n]$ the N_{GOP}-dimensional random variable

$$\mathbf{x}[n] \overset{\mathrm{def}}{=} \left[x[n \cdot N_{\mathrm{GOP}}], \cdots x[n \cdot N_{\mathrm{GOP}} + N_{\mathrm{GOP}} - 1]\right]^{\mathrm{T}} \tag{4.1}$$

representing the sizes, in bits, of the N_{GOP} frames constituting the n-th GOP of the coded video sequence. The transmission of a GOP is modeled as a realization of the random variable $\mathbf{x}[n]$, observed at the output of a first-order homogeneous Markov chain. The set of states of the chain Λ_s comprises as many states as the number of different GOP kinds, *i.e.* $N_s = L^2$, among which we recognize L states $R_i, i = 0, \cdots N-1$ representing GOPs emitted during streaming at an average rate r_i, and $L^2 - L$ states $S_{i,j}, i, j = 0, \cdots N-1, i \neq j$ representing GOPs emitted when an event of bitstream switching occurs.

The chain is described by the $N_s \times N_s$ transition matrix Π, whose generic element $\pi_{\lambda\mu}$ represents the transition probability from the state λ to the state μ. Let us observe that, due to decoding constraints on consecutive GOPs emitted by a real video streaming source, not all the transitions are admitted in the HMP model. Therefore, several elements of Π are equal to 0. An example of the model for $L = 2$, $N_s = 4$ is shown in Fig.4.1.

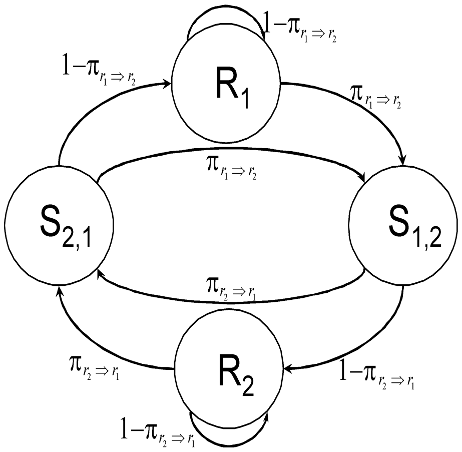

Figure 4.1: Markov chain representing an H.264 video source operating at $L = 2$ different streaming rates. The number of states of the Markov chain is $N_s = L^2 = 4$. The corresponding state transition probabilities are reported in Tab.4.1.

The statistical characteristics of the observed variable $\mathbf{x}[n]$ depend on the actual state $\lambda_n = \lambda$ of the Markov chain; specifically, we assume for $\mathbf{x}[n]$ the

From \ To	R_1	R_2	S_{12}	S_{21}
R_1	$1-\pi_{r_1 \Rightarrow r_2}$	0	$\pi_{r_1 \Rightarrow r_2}$	0
R_2	0	$1-\pi_{r_2 \Rightarrow r_1}$	0	$\pi_{r_2 \Rightarrow r_1}$
S_{12}	0	$1-\pi_{r_2 \Rightarrow r_1}$	0	$\pi_{r_2 \Rightarrow r_1}$
S_{21}	$1-\pi_{r_1 \Rightarrow r_2}$	0	$\pi_{r_1 \Rightarrow r_2}$	0

Table 4.1: State transition probabilities corresponding to the $N_s = 4$ markov chain in Fig.4.1.

following generation model:

$$\mathbf{x}[n] = \Sigma_\lambda \mathbf{e}[n] + \mathbf{c}_\lambda \tag{4.2}$$

where $\mathbf{e}[n]$ is a realization of a white, Gaussian, N_{GOP}-dimensional random process independent on the sequence of states λ_n, with

$$\begin{aligned} E\{\mathbf{e}[n]\} &= \mathbf{0} \\ E\{\mathbf{e}[n]\mathbf{e}[n-m]^{\mathrm{T}}\} &= I \cdot \delta[m] \end{aligned}$$

From the model (4.2), it stems out that the observation $\mathbf{x}[n]$ conditioned to the state $\lambda_n = \lambda$ is a normal random variable whose mean and covariance matrix depend on the actual state $\lambda_n = \lambda$, *i.e.*

$$p(\mathbf{x}[n]|\lambda_n = \lambda) = \mathcal{N}\left(\mathbf{x}[n], \mathbf{c}_\lambda, \Sigma_\lambda \Sigma_\lambda^{\mathrm{T}}\right) \tag{4.3}$$

The model identified by (4.3) belongs to the class of Gaussian Mixture (GM) processes with Markov dependence (20). The GM-HMP, widely studied in (22; 35; 36), has found application in automatic speech recognition; its application to video source modeling is novel under different respects. The first underlying assumption in (4.2) is that while inter-frame correlation within one GOP is taken into account by the matrix Σ_λ, inter-frame correlation between different GOPs is taken into account by the Markov chain structure; this assumption represents a novelty with respect to the literature, where AR models are rather employed (13; 16). Moreover, the frame size pdf is assumed to be normally distributed, whereas several works in literature assume heavily tailed pdfs such as Gamma or Beta distribution. Under the aforementioned modeling assumptions, we have derived the EM algorithm for estimation of the HMP parameters; in the following, we will show that the HMP driven by the parameters estimated by means of the EM algorithm accurately captures various statistical characteristics of H.264 coded video data.

Ergodicity conditions for a GM-HMP reside only on the properties of the hidden Markov chain: if the chain is stationary, irreducible and aperiodic, the model is ergodic (20). It is straightforward to verify that the Markov chain modeling the GOP sequence has indeed the properties listed above, and therefore the resulting GM-HMP source model is ergodic.

Let us denote by $p_\lambda, \lambda = 1, \cdots, N_s$, the limit state probabilities of the Markov chain. The mean vector and covariance matrix of the variate $\mathbf{x}[n]$ in (4.2), defined as:

$$\begin{aligned} &\mu_{\mathbf{x}} \overset{\text{def}}{=} E\left\{\mathbf{x}[n]\right\} \\ &R_{\mathbf{x}}\left[m\right] \overset{\text{def}}{=} E\{\mathbf{x}[n]\mathbf{x}[n-m]^T\} \end{aligned} \tag{4.4}$$

take the following forms:

$$\begin{aligned} \mu_{\mathbf{x}} &= \sum_{\lambda} p_\lambda \mathbf{c}_\lambda \\ R_{\mathbf{x}}\left[m\right] &= \sum_{\lambda_1}\sum_{\lambda_2} p_{\lambda_1} \|\Pi^m\|_{\lambda_1\lambda_2} \cdot \mathbf{c}_{\lambda_2}\mathbf{c}_{\lambda_1}^{\mathrm{T}} \\ &\quad + \delta[m]\sum_{\lambda} p_\lambda \Sigma_\lambda \Sigma_\lambda^{\mathrm{T}}. \end{aligned} \tag{4.5}$$

The compact model in (4.2) describes vectorial random variables modeling the N_{GOP}-uple generated by a video source during the encoding of a GOP, that is it refers to the GOP layer. From the HMP model in (4.2) several statistics referring to individual frame size can be inferred, a few examples of them being reported in A. In the Section devoted to the experimental results, we will show that the HMP in (4.2) provides a tight statistical model of the frame size sequence measured on real H.264 encoded video sequences.

Thanks to the model compactness, the behavior of the HMP in (4.2) is fully determined by the model parameter set

$$\Theta \overset{\text{def}}{=} \{\Pi, \Sigma_1, \ldots, \Sigma_{N_s}, \mathbf{c}_1, \ldots, \mathbf{c}_{N_s}\}$$

The aim of the modeling procedure is to design a synthetic source statistically similar to a real video source; then, the HMP shall be driven by a suitably designed parameter set. A basic design approach relies on observing a possibly short trace of video traffic and adjusting the model parameters to best describe the real data according to specifically selected criteria. In the following Section, we devise a parameter set estimation procedure based on the EM algorithm.

4.3 HMP Parameter Set Estimation

The EM algorithm, originally developed by Dempster, Laird and Rubin (18), allows to estimate the parameters of a HMP by local maximization of the likelihood function, and it has been applied in several fields of research, including image classification (4), image restoration (66), image deconvolution (6); recently, the EM algorithm has been employed to estimate the parameters of a switching Markov AR process modeling surface electromyographic signals (9).

The EM algorithm iteratively alternates two steps: i) evaluating an auxiliary log-likelihood function of the observations given a previous estimate of the unknown parameter set; ii) maximizing the so found auxiliary function to evaluate a new parameter set estimate. To outline the EM algorithm steps in detail, let us compactly denote by

$$\mathbf{x}_{0,N} \overset{\text{def}}{=} \{\mathbf{x}[n], n = 0 \cdots N-1\}$$

the observation sequence constituted by N consecutive video GOPs; moreover, let $\lambda_{0,N} \overset{\text{def}}{=} \{\lambda_n, n = 0 \cdots N-1\}$ denote the unknown underlying state sequence corresponding to the observed set $\mathbf{x}_{0,N}$.

Given the k-th estimated parameter set $\Theta^{(k)}$, the expectation step (E-step) computes the auxiliary likelihood function

$$Q(\Theta, \Theta^{(k)}) \overset{\text{def}}{=} \sum_{\lambda_{0,N}} \log\Big(p\left(\lambda_{0,N}, \mathbf{x}_{0,N}; \Theta\right)\Big) p\Big(\lambda_{0,N} | \mathbf{x}_{0,N}; \Theta^{(k)}\Big)$$

The maximization step (M-step) maximizes the auxiliary likelihood function $Q(\Theta, \Theta^{(k)})$ with respect to the unknown parameter set Θ so as to provide the $(k+1)$-th step estimate:

$$\Theta^{(k+1)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(k)}). \tag{4.6}$$

The algorithm stops when the parameter set $\Theta^{(k)}$ quits changing according to a suitable distance measure or when some different stop criterion is met.

The maximization problem is solved resorting to standard constrained optimization techniques, and the solution is expressed in terms of the so-called *re-estimation formulas*, which relate the maximization solution, *i.e.* the new estimate $\Theta^{(k+1)}$, to the observations $\mathbf{x}_{0,N}$ and to the conditional probabilities of the unknown state sequence $\lambda_{0,N}$ given the observations $\mathbf{x}_{0,N}$ and the previous parameter set estimate $\Theta^{(k)}$. For the model described in Sect.4.2, in which the observation conditional density in the state λ_n is Gaussian, the re-estimation formulas are evaluated as follows (20):

$$\mathbf{c}_\lambda^{(k+1)} = \left(\sum_{n=0}^{N-1} \gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}) \right)^{-1} \sum_{n=0}^{N-1} \gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)})\ \mathbf{x}[n] \tag{4.7}$$

$$\begin{aligned} \Sigma_\lambda^{(k+1)} \left(\Sigma_\lambda^{(k+1)} \right)^{\mathrm{T}} = & \left(\sum_{n=0}^{N-1} \gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}) \right)^{-1} \\ & \sum_{n=0}^{N-1} \gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)})(\mathbf{x}[n] - \mathbf{c}_\lambda^{(k+1)})(\mathbf{x}[n] - \mathbf{c}_\lambda^{(k+1)})^{\mathrm{T}} \end{aligned} \tag{4.8}$$

$$\pi_{\lambda,\mu}^{(k+1)} = \left(\sum_{n=0}^{N-2} \gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}) \right)^{-1} \sum_{n=0}^{N-2} \xi_n(\lambda, \mu; \mathbf{x}_{0,N}, \Theta^{(k)}) \tag{4.9}$$

where we have adopted the compact notation:

$$\gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}) \overset{\text{def}}{=} p\left(\lambda_n = \lambda |\ \mathbf{x}_{0,N}, \Theta^{(k)} \right)$$
$$\xi_n(\lambda, \mu; \mathbf{x}_{0,N}, \Theta^{(k)}) \overset{\text{def}}{=} p\left(\lambda_n = \lambda, \lambda_{n+1} = \mu |\ \mathbf{x}_{0,N}, \Theta^{(k)} \right)$$

Using the algorithm described in (20), and here reported in Tab.4.2 for reader's convenience, the terms $\gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)})$, $\xi_n(\lambda, \mu; \mathbf{x}_{0,N}, \Theta^{(k)})$, which represent the state conditional probabilities given the whole set of N observations $\mathbf{x}_{0,N}$, can be recursively evaluated in terms of the conditional probabilities given a smaller subset of n observations, that is

$$\alpha_n\left(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)} \right) \overset{\text{def}}{=} p\left(\lambda_n = \lambda | \mathbf{x}_{0,n}, \Theta^{(k)} \right)$$

The probability $\alpha_n\left(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)} \right)$ depends on the following observations' conditional pdf given the previously estimated $\Theta^{(k)}$:

$$f_n\left(\mathbf{x}_{0,N}; \lambda, \Theta^{(k)} \right) \overset{\text{def}}{=} p\left(\mathbf{x}[n] | \lambda_n = \lambda, \Theta^{(k)} \right).$$

Different recursions for estimating $\alpha_n\left(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}\right)$ appear in literature; the results shown in Sect.4.4 have been obtained by adopting the forward stable recursion provided in (19; 40), and summarized in Tab.4.3.

$$\gamma_{N-1}(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}) = \alpha_{N-1}\left(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}\right)$$

$$\begin{aligned}
\xi_n(\lambda, \mu; \mathbf{x}_{0,N}, \Theta^{(k)}) &= \pi^{(k)}_{\mu\lambda}\gamma_{n+1}(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)})\alpha_n\left(\mu; \mathbf{x}_{0,N}, \Theta^{(k)}\right) \cdot \\
&\quad \sum_{\mu} \pi^{(k)}_{\mu\lambda}\alpha_n\left(\mu; \mathbf{x}_{0,N}, \Theta^{(k)}\right)^{-1} \\
\gamma_n(\lambda; \mathbf{x}_{0,N}, \Theta^{(k)}) &= \sum_{\mu} \xi_n(\lambda, \mu; \mathbf{x}_{0,N}, \Theta^{(k)}) \\
n &= N-2, \ldots, 0
\end{aligned}$$

Table 4.2: Algorithm for stable estimation of the observation conditional probabilities $\gamma_n(\lambda)$ and of the transition probabilities $\xi_n(\mu, \lambda)$.

Then, based on the aforementioned Gaussian Mixture HMP, we have analytically developed the parameter estimation procedure in the EM framework; the resulting locally ML parameter estimation algorithm, to the best of the authors' knowledge, appears here for the first time in the context of video source modeling, where parameter estimation is typically performed resorting to heuristic criteria.

4.4 Numerical Simulations

In this Section, we first present numerical simulations results that illustrate the convergence of the EM algorithm to the true HMP model parameters in a realistic scenario of limited *a priori* information and reduced number of observations.

After having discussed the HMP parameter set estimation algorithm, we will show that the HMP model using estimated parameters achieves satisfying performance in capturing the significant statistical characteristics of the data encoded by a real H.264 source. In particular, Subsect. 4.4.1 is devoted to the discussion

$$\alpha_0\left(\lambda;\mathbf{x}_{0,N},\Theta^{(k)}\right)=f_0\left(\mathbf{x}_{0,N};\lambda,\Theta^{(k)}\right)\pi_\lambda^{(k)}\cdot\left[\sum_\mu \pi_\mu^{(k)} f_0\left(\mathbf{x}_{0,N};\mu,\Theta^{(k)}\right)\right]^{-1}$$

$$\alpha_n\left(\lambda;\mathbf{x}_{0,N},\Theta^{(k)}\right)= f_n\left(\mathbf{x}_{0,N};\lambda,\Theta^{(k)}\right)\sum_\mu \pi_{\mu\lambda}^{(k)}\alpha_{n-1}\left(\mu;\mathbf{x}_{0,N},\Theta^{(k)}\right)\cdot \left[\sum_\delta f_n\left(\mathbf{x}_{0,N};\delta,\Theta^{(k)}\right)\sum_\mu \pi_{\mu\delta}^{(k)}\alpha_{n-1}\left(\mu;\mathbf{x}_{0,N},\Theta^{(k)}\right)\right]^{-1}$$

$$n=1,\ldots,N-1$$

Table 4.3: Algorithm for stable recursion of the state conditional probabilities $\alpha_n\left(\lambda;\mathbf{x}_{0,N},\Theta^{(k)}\right)$.

of results of EM algorithm convergence, while Subsect. 4.5 is devoted to assess the HMP model performance in mimicking a selected set of statistics observed on video data encoded by a real H.264 source performing bitstream switching.

4.4.1 EM algorithm convergence

Since the likelihood function of an HMP has, in general, several local maxima, the convergence of the EM algorithm is severely affected by the initial parameter set estimate $\Theta^{(0)}$. The EM algorithm convergence has been verified on HMPs operating at different average bit-rates. Here, we present the results obtained by assigning $N_{\text{GOP}}=10$ and nominal rates $r_1=20$ kbps, $r_2=50$ kbps; this choice is motivated by the fact that these two nominal rates are close enough to stress the EM algorithm in identifying the bitstreams; similar results have been obtained at different nominal bit-rates. The state dependent parameters $\mathbf{c}_\lambda$ and Σ_λ adopted in the simulations are reported in Tab.4.4, where it can be observed that all the P frames pertaining to the same state have been assigned equal first and second order moments. The HMP transition probabilities $\pi_{r_1\Rightarrow r_2}$ and $\pi_{r_2\Rightarrow r_1}$ have been set to 0.4 and 0.7. The algorithm operates after the observation of a sequence of

GOPs $\mathbf{x}[0], \cdots, \mathbf{x}[N-1]$, with $N = 25$.

State	$\mathbf{c}_\lambda$	Σ_λ
R_1	$[8413\ 1335 \cdots 1335]^{\mathrm{T}}$	$\mathrm{diag}\,(802, 335, \cdots, 335)$
R_2	$[23514\ 3377 \cdots 3377]^{\mathrm{T}}$	$\mathrm{diag}\,(1331, 675, \cdots, 675)$
S_{12}	$[31911\ 3377 \cdots 3377]^{\mathrm{T}}$	$\mathrm{diag}\,(970, 675, \cdots, 675)$
S_{21}	$[11230\ 1335 \cdots 1335]^{\mathrm{T}}$	$\mathrm{diag}\,(648, 335, \cdots, 335)$

Table 4.4: Parameter $\mathbf{c}_\lambda$ and Σ_λ estimated by the EM algorithm.

EM algorithm needs a suitable initialization of the parameter set to begin the iterations: the choice of this initial point is crucial in order to reach the global maximum of the likelihood function (18). To start in a point enabling the convergence to the global maximum, while still exploiting the minimal *a priori* knowledge about the nominal bit-rate and the transition matrix structure, we set the elements of the vector $\mathbf{c}_\lambda$ and the matrix Σ_λ as a function of the nominal streaming bit-rates r_1, r_2. Namely, with reference to the Markov chain in Fig.4.1, we initialize the elements of the vector $\mathbf{c}_\lambda$ with a constant value related to the nominal rate r_i[1]:

$$\|\mathbf{c}_\lambda^{(0)}\|_i = r_\lambda \cdot T_{GOP}/N_{\mathrm{GOP}},\ i = 0, \ldots, N_{\mathrm{GOP}} - 1 \tag{4.10}$$

being $r_\lambda = r_1$ for the states R_1, S_{21} and $r_\lambda = r_2$ for the states R_2, S_{12}, and being T_{GOP} the GOP period. The matrix $\Sigma_\lambda^{(0)}$ is assumed to be proportional to the identity matrix and trace equal to the average GOP bit budget, namely:

$$\Sigma_\lambda^{(0)} = \left(\frac{1}{L} \sum_{\lambda=1}^{L} r_\lambda \cdot T_{GOP}/N_{\mathrm{GOP}} \right) \cdot I \tag{4.11}$$

Finally, as far as the transition matrix Π is concerned, any element corresponding to bitstream switching that is not allowed by the decoding constraints is set to zero, while the non-zero transition probabilities are assumed to be all equal.

The iterative EM algorithm is applied in accordance to the general outline given in Sect. 4.3; furthermore, at the k-th iteration of the algorithm, we impose a diagonal structure to Σ_λ by setting equal to zero the extra diagonal entries and

[1] Let us observe that, in both cases, these constants exceed the true mean values on P-frames and are less than the true mean value on SP-frame.

by straightforwardly evaluating its diagonal elements, representing the standard deviation of the frames pertaining to a specific GOP kind, as follows:

$$\|\Sigma_\lambda^{(k+1)}\|_{ii} = \sqrt{\frac{\sum_{n=0}^{N-1} \gamma_n(\lambda)\left(\|\mathbf{x}[n]\|_i - \|\mathbf{c}_\lambda^{(k+1)}\|_i\right)^2}{\sum_{n=0}^{N-1}\gamma_n(\lambda)}},$$

$$i = 0, \cdots N_{\mathrm{GOP}} - 1$$

Convergence of the EM algorithm is illustrated in Figs.4.2-4.8, where estimated and true parameters' values are plotted versus the iteration number. Since the algorithm estimates N_{GOP} elements for each vector $\mathbf{c}_\lambda$, and N_{GOP} for each matrix Σ_λ matrices, while only two elements are needed each vector and matrix, for SP frames we plot the following statistics:

$$\|\mathbf{c}_\lambda^{(k)}\|_0, \|\Sigma_\lambda^{(k)}\|_{00}, \ \ \lambda = 1, \ldots, N_S - 1$$

while for P frames we plot

$$\begin{aligned}\overline{\|\mathbf{c}_\lambda^{(k)}\|_i} &= \frac{1}{N_{\mathrm{GOP}}-1}\sum_{i=1}^{N_{\mathrm{GOP}}-1}\|\mathbf{c}_\lambda^{(k)}\|_i \\ \overline{\|\Sigma_\lambda^{(k)}\|_{ii}} &= \frac{1}{N_{\mathrm{GOP}}-1}\sum_{i=1}^{N_{\mathrm{GOP}}-1}\|\Sigma_\lambda^{(k)}\|_{ii}\end{aligned} \tag{4.12}$$

being k the iteration number.

All the parameters converge to their true values after very few iterations of the EM algorithm, although mean values and standard deviations of P frames in the switching and non-switching GOPs are very similar; moreover, convergence is quite robust under different initialization, and only underestimation of the standard deviation parameters appearing in the matrix Σ_λ should be avoided, since it directly affects the state sequence estimation implicitly performed by the EM algorithm.

We observe that the compact form of the parameter space has beneficial effects on the EM algorithm convergence, because it allows convergence after very few iterations also in presence of a limited number of observed GOPs. On the other hand, we will show that a satisfying approximation of the statistics of real traffic is attained both in terms of autocorrelation and of corresponding loss and even though the inter-frame intra-GOP correlation is neglected.

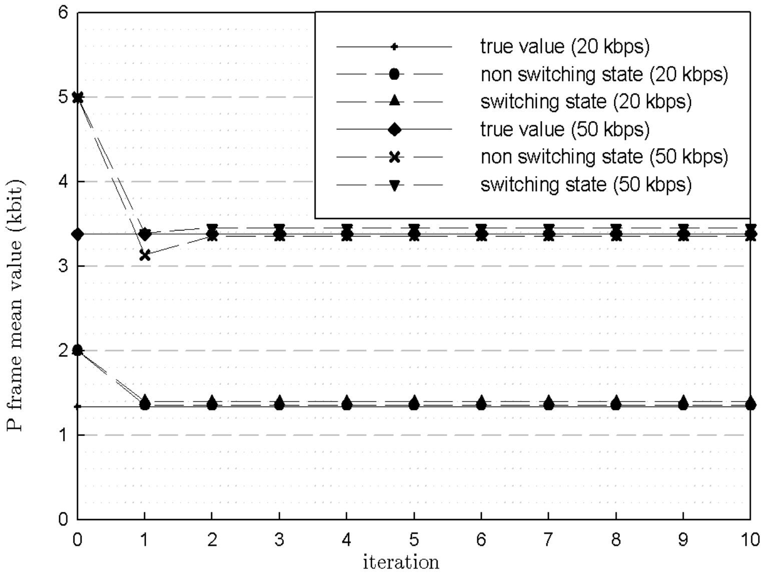

Figure 4.2: P frame mean value: true value at 20 kbps (plus), true value at 50 kbps (diamond), estimation from non-switching state at 20 kbps (circle), estimation from non-switching state at 50 kbps (cross), estimation from switching state at 20 kbps (triangle-up), estimation from switching state at 50 kbps (triangle-down).

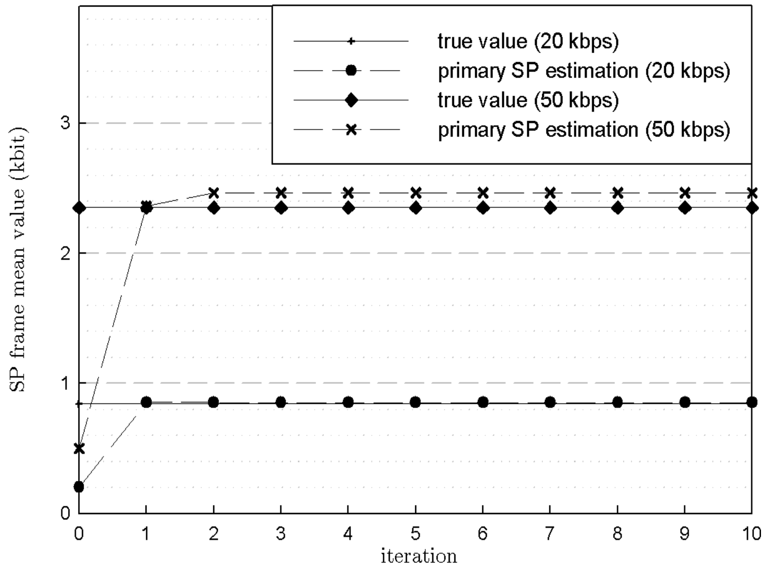

Figure 4.3: Primary SP frame mean value: true value at 20 kbps (plus), true value at 50 kbps (diamond), estimation at 20 kbps (circle), estimation at 50 kbps (cross).

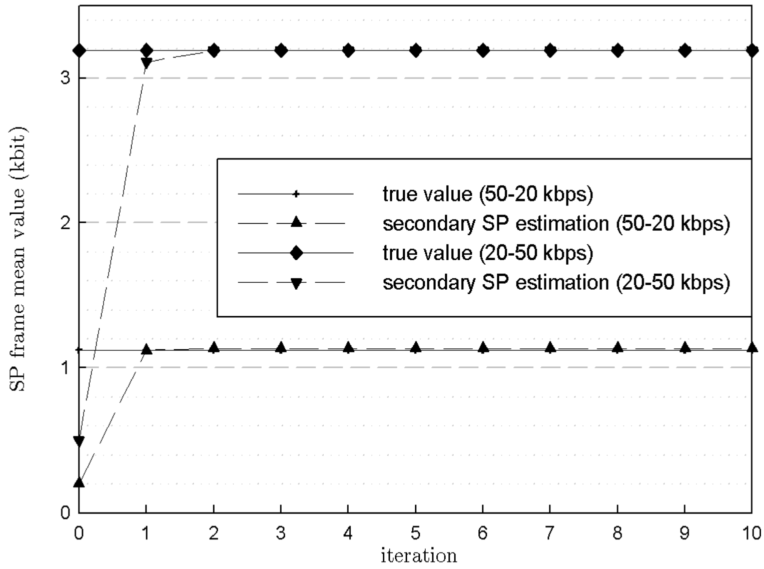

Figure 4.4: Secondary SP frame mean value: true value at 50-20 kbps (plus), true value at 20-50 kbps (diamond), estimation at 50-20 kbps (triangle-up), estimation at 20-50 kbps (triangle-down).

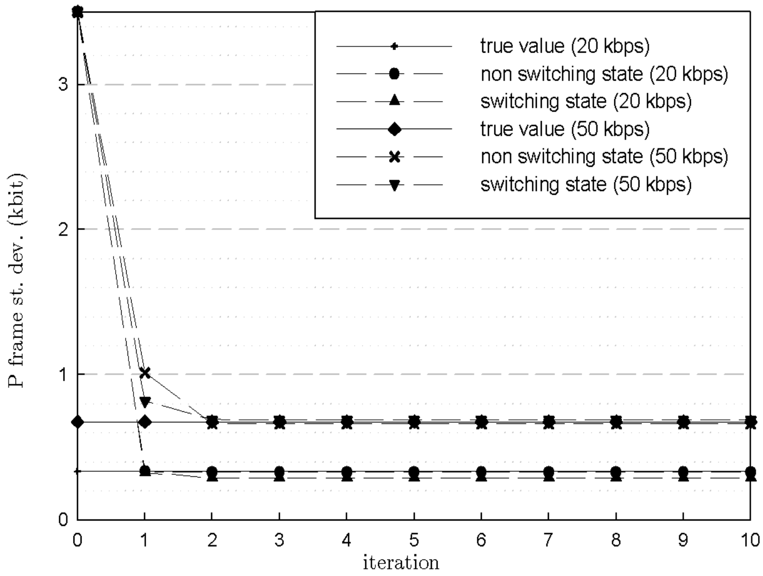

Figure 4.5: P frame standard deviation: true value at 20 kbps (plus), true value at 50 kbps (diamond), estimation from non-switching state at 20 kbps (circle), estimation from non-switching state at 50 kbps (cross), estimation from switching state at 20 kbps (triangle-up), estimation from switching state at 50 kbps (triangle-down).

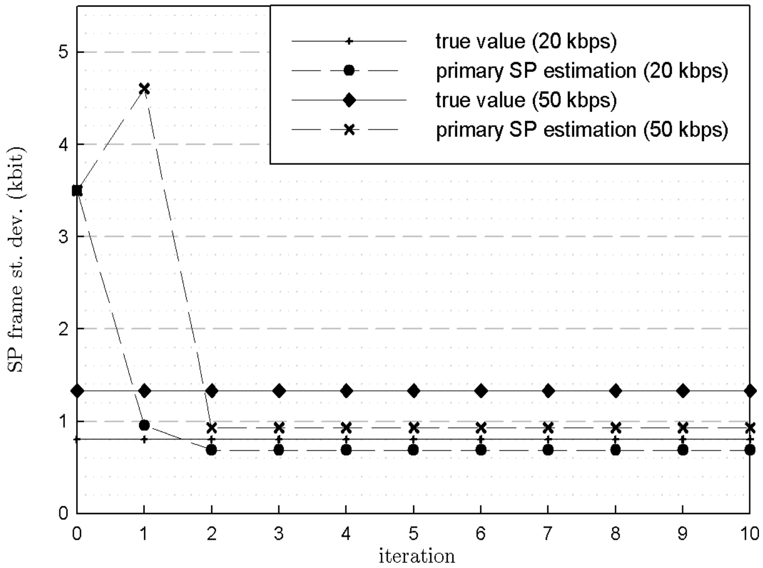

Figure 4.6: Primary SP frame standard deviation: Primary SP frame mean value: true value at 20 kbps (plus), true value at 50 kbps (diamond), estimation at 20 kbps (circle), estimation at 50 kbps (cross).

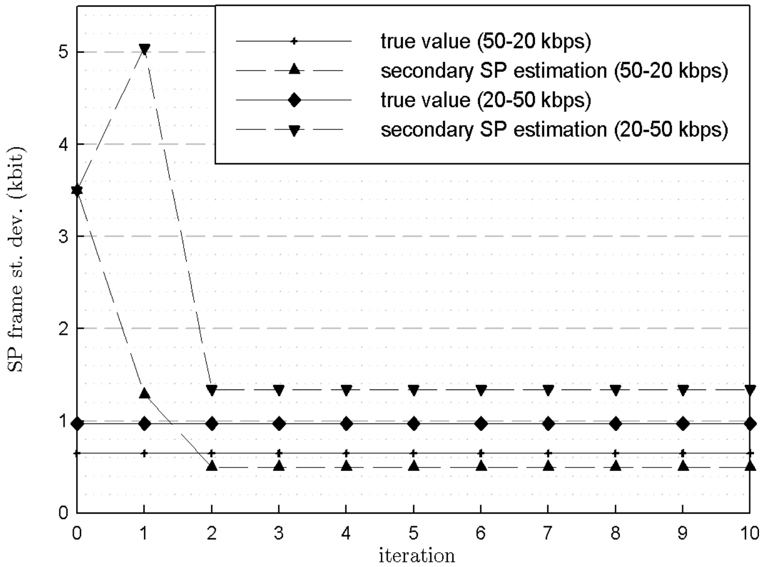

Figure 4.7: Secondary SP frame standard deviation: true value at 50-20 kbps (plus), true value at 20-50 kbps (diamond), estimation at 50-20 kbps (triangle-up), estimation at 20-50 kbps (triangle-down).

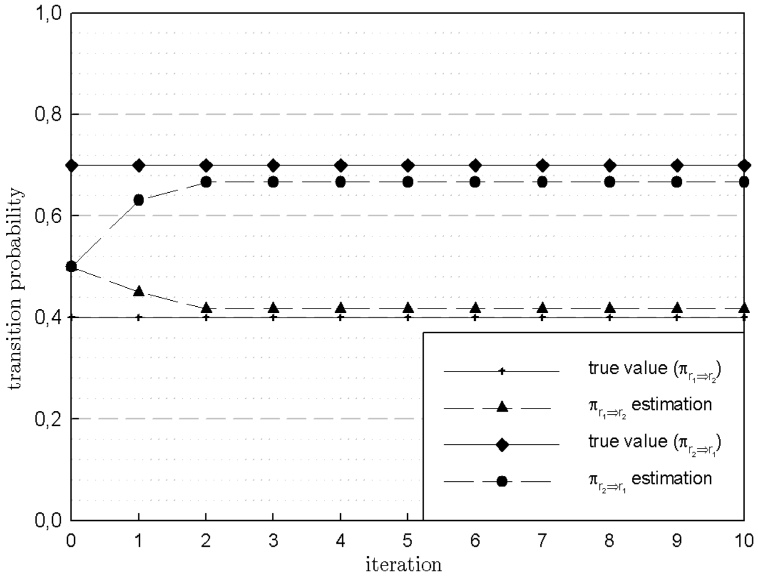

Figure 4.8: Transition probability estimation: true value $\pi_{r1 \Rightarrow r2}$ (plus), true value $\pi_{r2 \Rightarrow r1}$ (diamond), estimation $\pi_{r1 \Rightarrow r2}$ (triangle-up), estimation $\pi_{r2 \Rightarrow r1}$ (circle).

4.5 HMP model performances

After having discussed the convergence of the EM algorithm, in this Section we evaluate the performance of the herein analyzed Markov chain in modeling the traffic generated by a real video source. We consider as case-study a transmission into EDGE channel.

4.5.1 EDGE Channel Model

The EDGE channel model is first introduced in (7) and used in (53), (29),(38) The authors characterize the EDGE channel by a layered description. Two layers are presented: in the top level, users are divided into two groups, group 1 representing bad locations group and group 2 good locations group. These two groups summarize the possible conditions that may happen due to intereference, shadowing, low/high received signal power, etc. Not all the modulation/channel scheme codings are possible in each group, only a subset characterized by the states of a Markov chain, *i.e.* in group 1 users vary their channel/modulation scheme according to a 3-state Markov chain of resp. 11.2 kbps, 14.8 kbps, and 17.6 kbps. In group 2, a 2-state Markov chain of 49.5 kbps and 59.2 kbps is assumed. In Fig.4.9 the overall scheme is presented. Tab.4.5 shows parameter values according to (29) and Tab.4.6 according to (53). Group changes happen independently every second, but (53) assumes changes every 20 seconds. The former hypothesis is assumed in the following. State changes happen every tenth of second according to the group relative Markov chain.

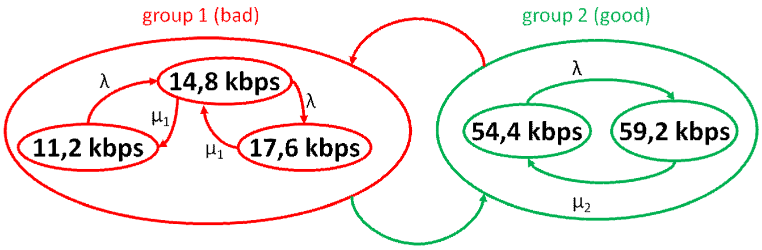

Figure 4.9: EDGE channel model.

Users	p_{g1}	λ	μ_1	μ_2
2	0.029	0.1	0.069	0.015
4	0.062	0.1	0.069	0.04
6	0.126	0.1	0.076	0.082
8	0.235	0.1	0.096	0.176
10	0.456	0.1	0.147	0.4
12	0.588	0.1	0.191	0.571
14	0.651	0.1	0.219	0.658

Table 4.5: Parameter values according to (29).

Users	p_{g1}	λ	μ_1	μ_2
2	0.07	0.3	0.055	0.05
8	0.36	0.3	0.094	0.3
15	0.73	0.3	0.27	0.59

Table 4.6: Parameter values according to (53).

4.5.2 Simulation settings

The sequence is encoded using a loose rate control (reference encoder rate control with the whole frame as basic unit, and SP frames encoded with QP minus 3 in order to have similar PSNR with respects to P frames) so as to have 30 kbps and 110 kbps of average bit-rate. The GOP structure is SP-P-P-P-P-P-P-P-P-P. A static assignment of two timeslots for the transmission is assumed, *i.e.* the rates in the Markov chains are doubled. The analysis are limited in the more frequent case of 14 users per cell (Jiang parameters). The server switches stream at every group change. Model is trained by observing the server stream output by means of the EM algorithm. Three different sequences are presented:

1. a QCIF compound sequence created by concatenating several test sequences (see Tab.4.7 for details) and decimated at 10 fps (total: 3750 frames);
2. the last 25 minutes of the FIFA World Cup final match, resized at QCIF and decimated at 10 fps (total: 12500 frames);
3. the entire movie "Indagine su un cittadino al di sopra di ogni sospetto"resized at QCIF and decimated at 10 fps (total: 55100 frames).

Sequence Name	total frames
akiyo	100
bridge close	2001
bridge far	2101
carphone	382
claire	494
coastguard	300
container	300
foreman	300
grandma	870
hall monitor	300
highway	2000
miss america	150
mobile	300
mother and daughter	300
news	300
salesman	449
silent	300
suzie	150
compound	11297

Table 4.7: Compound sequence composition.

4.6 Experimental Results

We have compared the autocorrelation function of the real sequence with the expected value of the model, the histogram with the model pdf and the 2dpdf (at lag N_{GOP} and lag 1). Figs.4.10-4.23 show results for the three sequences.

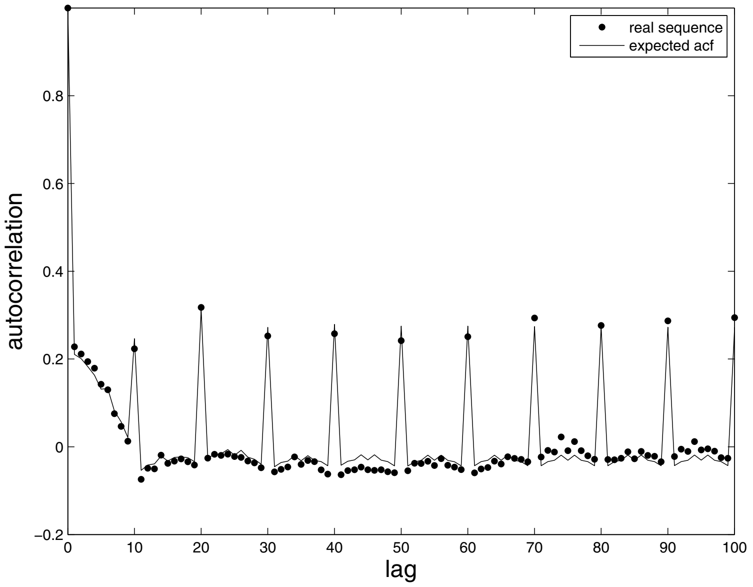

Figure 4.10: Autocorrelation for the compound sequence.

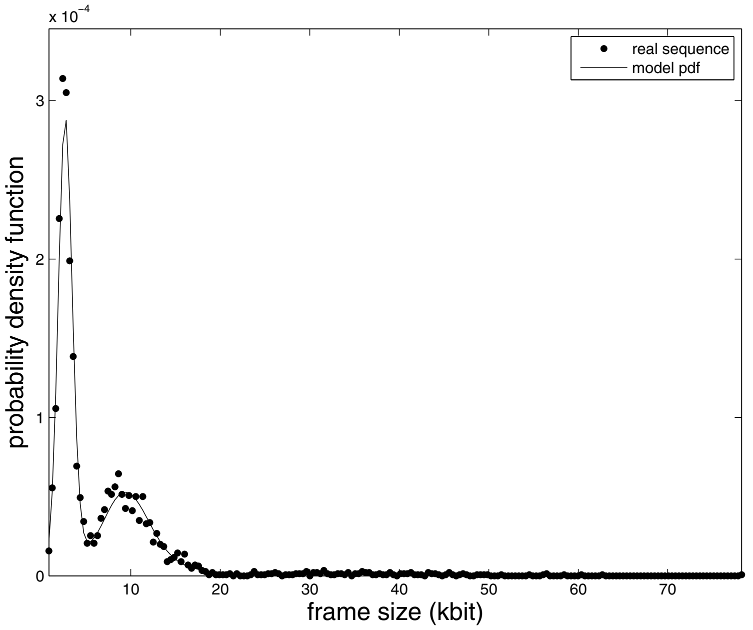

Figure 4.11: Histogram for the compound sequence.

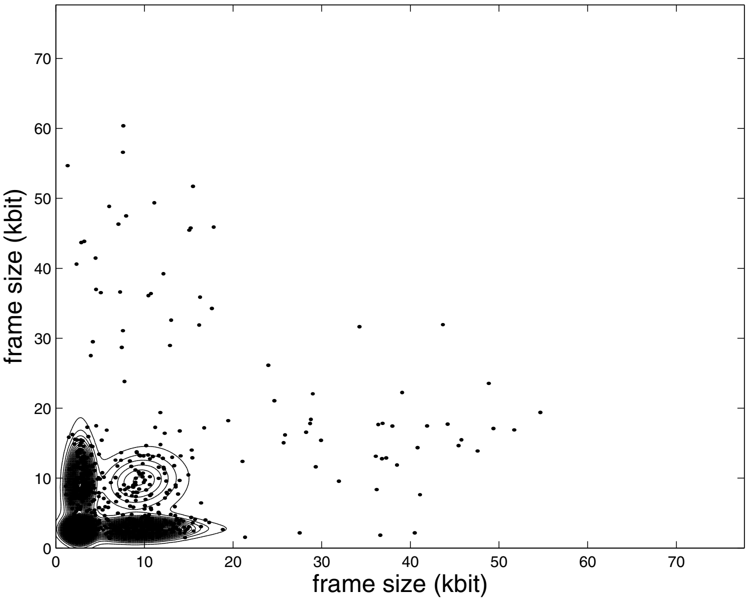

Figure 4.12: Joint pdf (N_{GOP} lag) for the compound sequence.

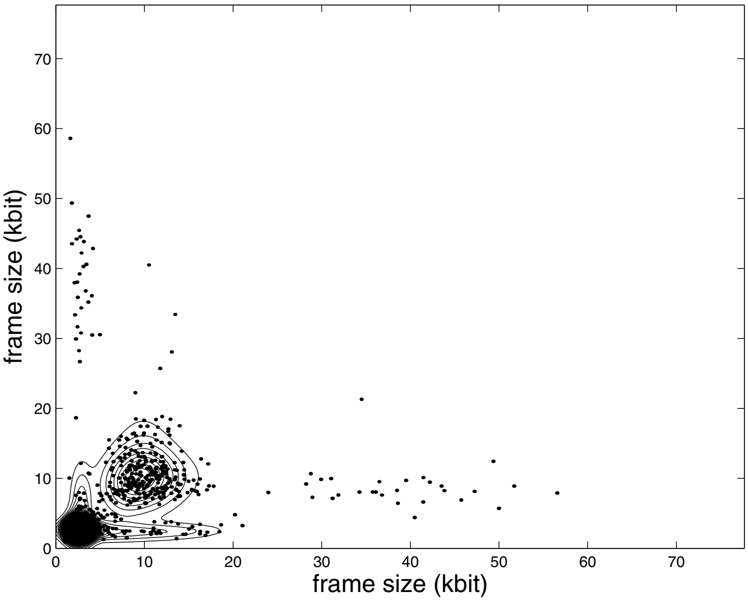

Figure 4.13: Joint pdf (1 lag) for the compound sequence.

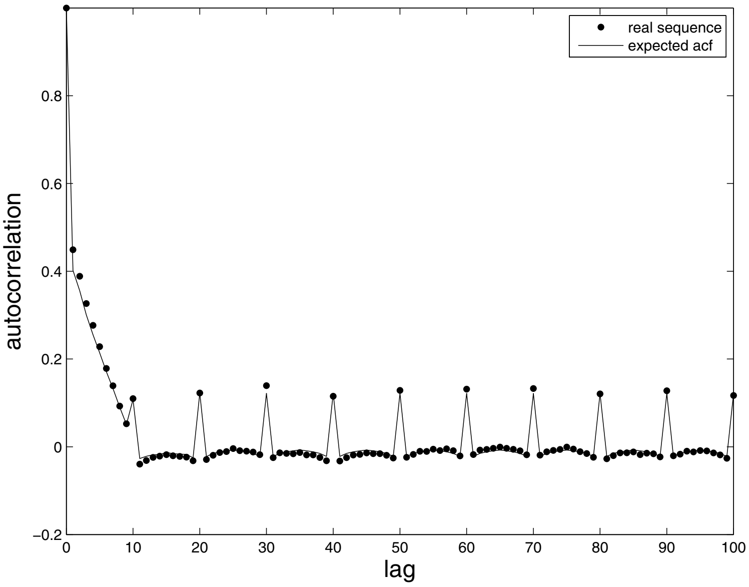

Figure 4.14: Autocorrelation for the indagine sequence.

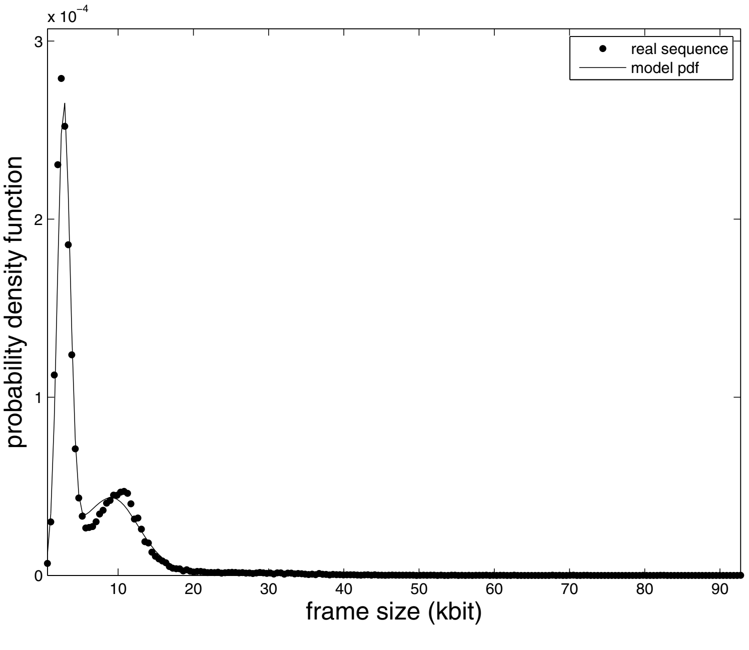

Figure 4.15: Histogram for the indagine sequence.

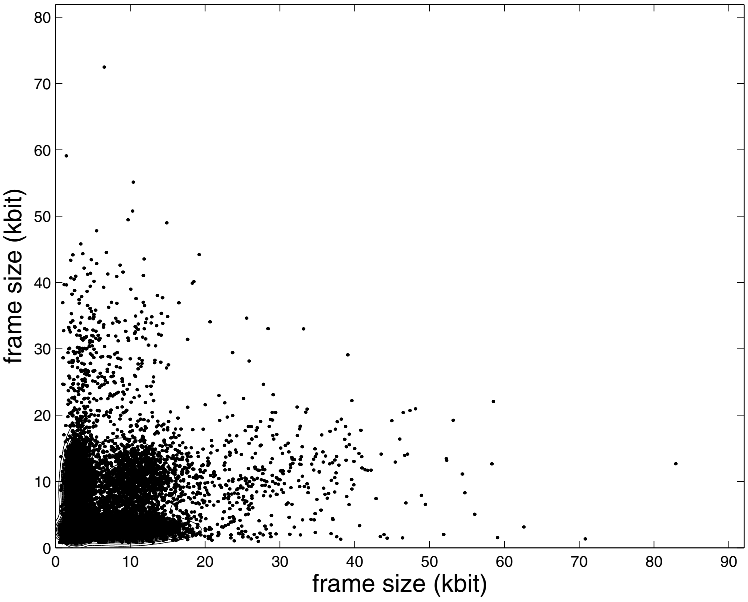

Figure 4.16: Joint pdf (N_{GOP} lag) for the indagine sequence.

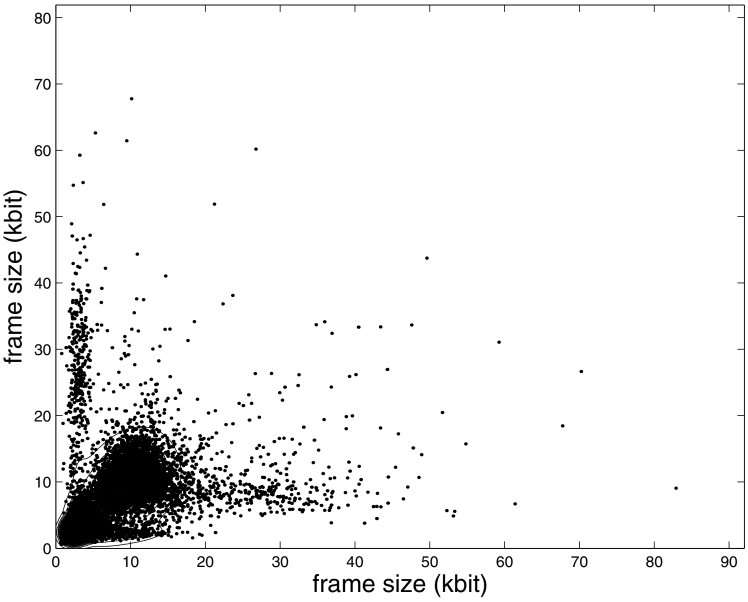

Figure 4.17: Joint pdf (1 lag) for the indagine sequence.

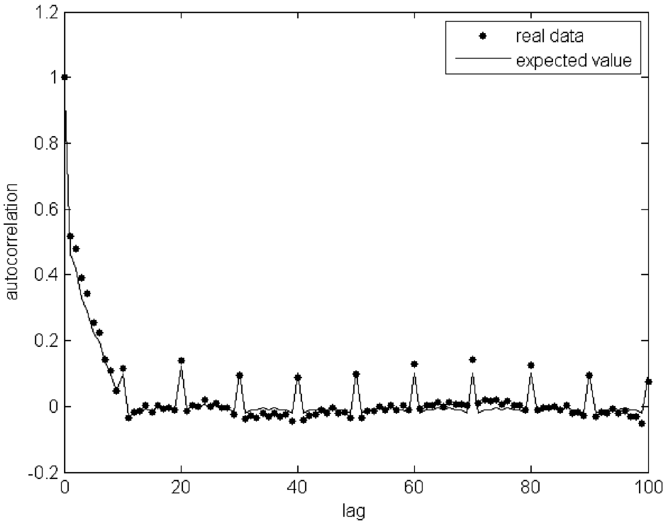

Figure 4.18: Autocorrelation for the world cup sequence.

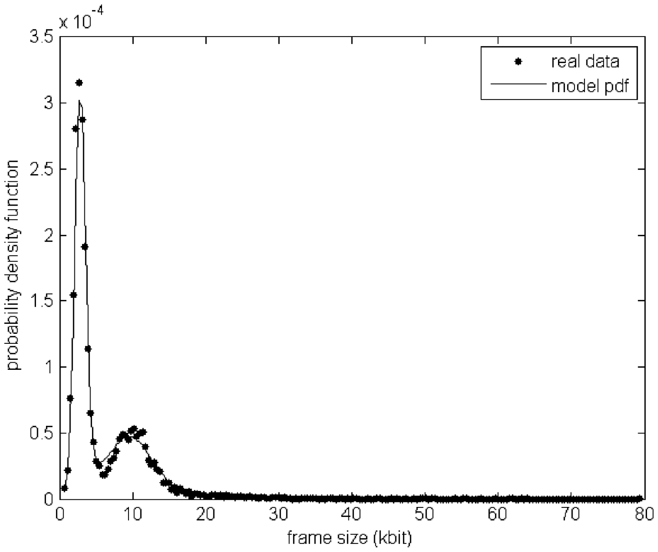

Figure 4.19: Histogram for the world cup sequence.

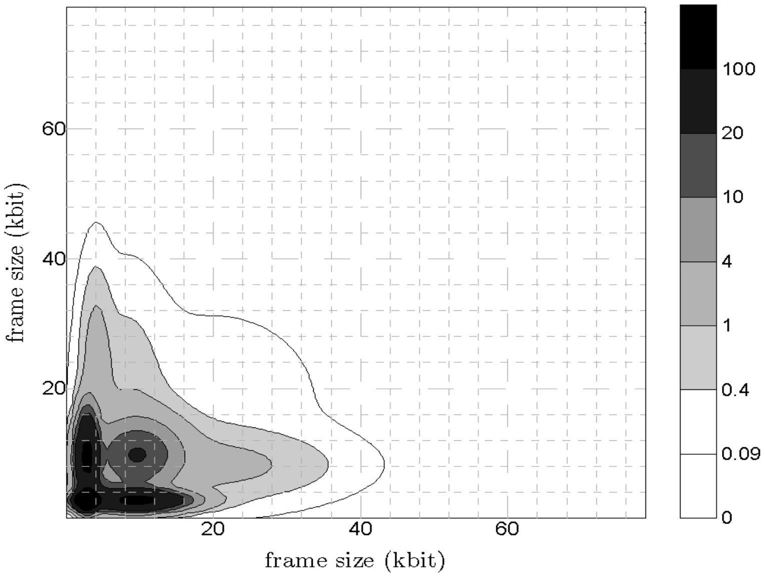

Figure 4.20: Joint pdf (N_{GOP} lag) for the world cup sequence.

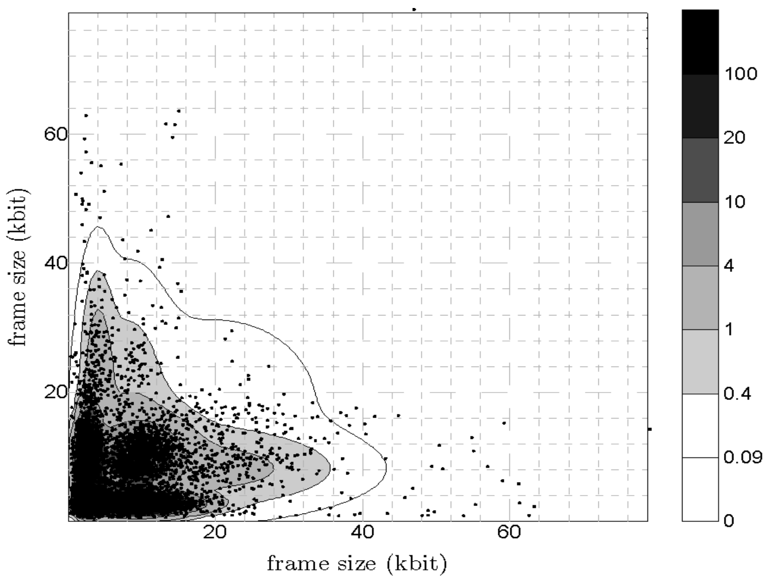

Figure 4.21: Joint pdf (N_{GOP} lag) for the world cup sequence.

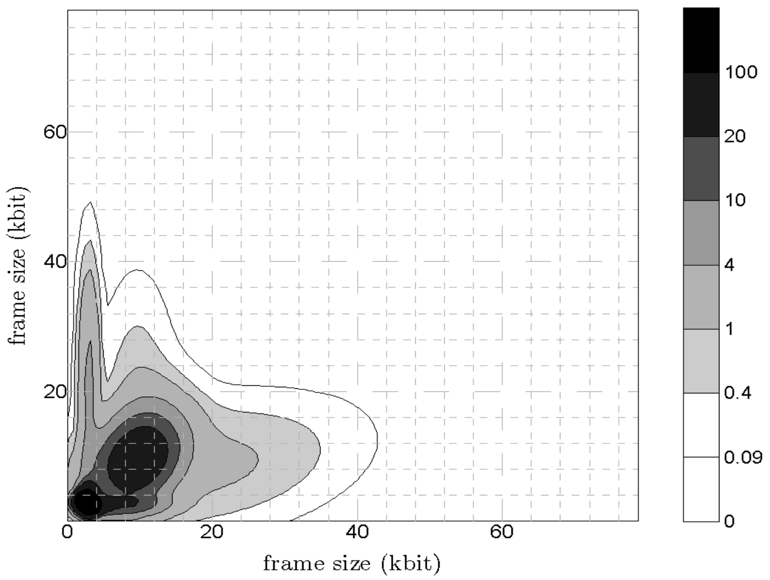

Figure 4.22: Joint pdf (1 lag) for the world cup sequence.

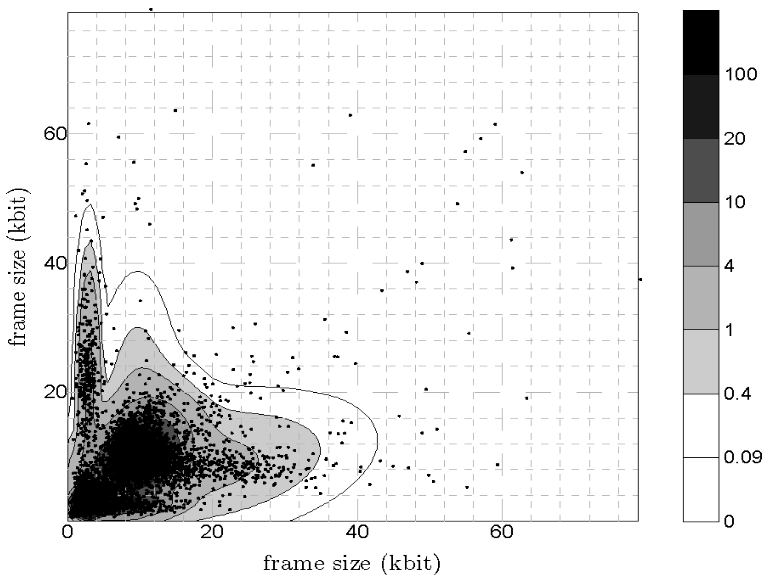

Figure 4.23: Joint pdf (1 lag) for the world cup sequence.

The previous analysis has shown that the HMP model mimics the statistical characteristics of the encoded video data. Then, the model can be exploited to predict the impact of a real video source traffic in terms of network load, by replacing real traces of a bitstream switching H.264 video source, whose measurement can be difficult and/or expensive, with synthetic traffic data generated by the HMP model. To verify this claim, we have evaluated the loss rate of a transmission buffer at whose input we have applied the H.264 coded sequence and a synthetic sequence $x[m]$ generated by the HMP source. The buffer continuously transmits at rate $r \in \{r_1, r_2\}$, using a first-in first-out policy. The processes of buffer filling and depleting are shown in Fig.5.2: the stepwise curve represents the incoming data, written in group of $\tilde{x}[m]$ or $x[m]$ bits; the straight lines represents the outgoing data, and the channel rate determines the straight lines slope.[1] The buffer output rate changes immediately according to the state of the EDGE channel and the server switches in correspondance of the group in the layered characterization of the channel. The $\tilde{x}[m]$ or $x[m]$ bits representing the m-th frame of the natural or synthetic sequence is stored in the buffer if and only if there is available space for it; otherwise, it is discarded. Figs.4.26,4.24,4.25 plot the frame loss rate versus the buffer size B, observed on the real source traffic $\tilde{x}[m]$ and on the HMP output $x[m]$. Therefore, we recognize that the HMP model well reproduces the characteristics of the data generated by the real video source from the point of view of the traffic load offered to the network.

4.7 Summary

The H.264 video coding standard introduces novel encoding tools intended to allow efficient rate adaptation in video streaming services. The tools enable fast bitstream switching by means of the syntactic element Switching Pictures. In this work a Hidden Markov Process, namely a Gaussian Mixture Markov Process, is employed to model the size of the frames encoded by a H.264 source possibly performing bit-rate adaptation using Switching Pictures. The model is compactly parameterized and estimation of the model parameter set has been performed via the EM algorithm. We have discussed the ability of the EM parameter esti-

[1] In more detail, the buffer is read by elementary units of $C = 384$ bits (corresponding to the 48 bytes payload of an ATM cell) at a rate of r_i/C cells per second.

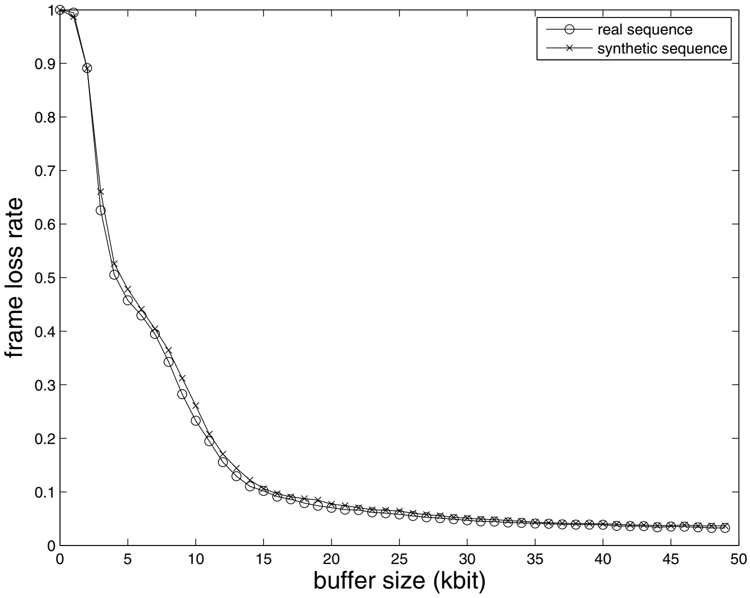

Figure 4.24: Frame loss rate comparison for compound sequence.

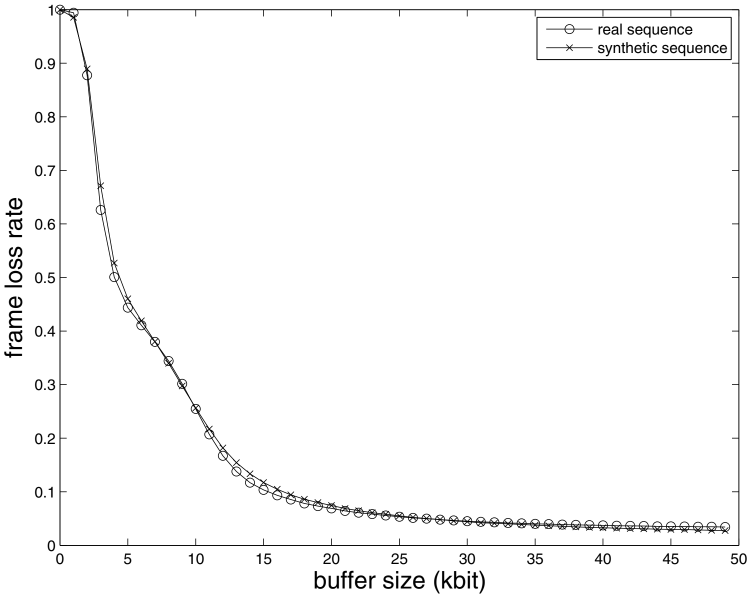

Figure 4.25: Frame loss rate comparison for indagine sequence.

mation algorithm to converge in presence of a small observation set and scarce *a priori* information. The model performance have been assessed by comparing

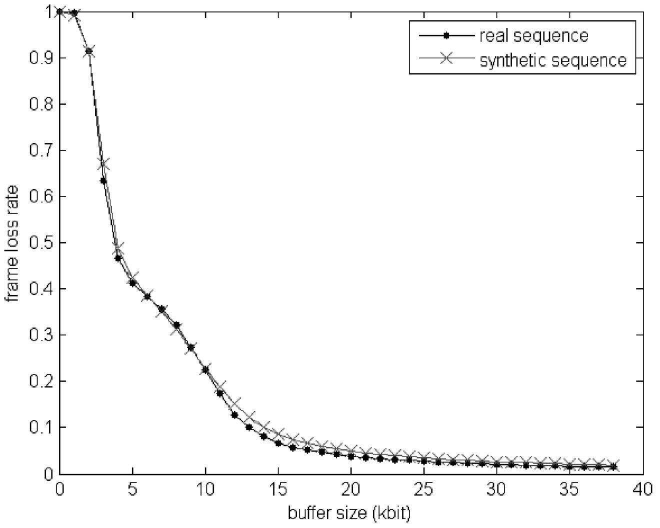

Figure 4.26: Frame loss rate comparison for sequence world cup.

first and second order HMP statistics with the corresponding sample statistics measured at the output of a real H.264 source. Furthermore, the sources have been compared in terms of the traffic offered to the network by evaluating the load of a transmission buffer. Numerical simulations show that the herein described Markovian model provides a compact and tight description of the behavior of a real video source possibly performing bitstream switching, and can be exploited for several purposes, ranging from optimization of network resource allocation to definition of call-admission control strategies, or to design of cross-layer optimized transmission algorithm.

Chapter 5

Improving User Experience: Multiview Video Coding

5.1 Introduction

Multiview Video Coding (MVC) is a new standard for the joint compression of correlated video sequences (views) representing the same scene recorded simultaneously by multiple cameras (8). The large amount of data and its high rate variability that are typical of MVC content necessitate accurate network dimensioning and resource allocation for successful deployment of multiview applications. Furthermore, the multiple encoding dependencies between the different views make allocating resources in an MVC system much more complex relative to single view scenarios.

Video traffic modeling is an active research area aimed at characterizing the behavior of compressed video content through statistical models. There is a substantial amount of related work on video traffic modeling ranging from applications in teleconferencing (67) to video streaming (14). In this regard, different stochastic models such as autoregressive processes (13), Transform Expanded Sample (TES) processes (44), and HMMs (15) have been considered. The proposed models are then typically applied to network dimensioning and provisioning, i.e., as a good aid for efficient and accurate allocation of network resources. For instance, one straightforward application of a video model is for generating a synthetic bitstream that is used afterwards to determine the proper size of a network buffer. Moreover, a video traffic model can also be used for deriving procedures for network call-admission-control (69).

Because of the recent nature of multiview applications, there are still no statistical characterizations of MVC compressed content. The present chapter provides the first stochastic model that characterizes the frame size sequence of an MVC compressed variable bit rate (VBR) multiview content. To this end, in Section 5.2, we design a non-stationarity HMM with a Poisson state duration distribution where the different states of the model represent different activity levels of the video source. Having a non-geometric state occupancy distribution allows us to more accurately model the scene activity duration in video content (26), while the specific choice of Poisson distribution allows our model to have the same complexity as a conventional HMM (20),(46). In Section 5.2, we also derive a numerically stable maximum likelihood procedure for estimating the parameters of the proposed model. Then, in Section 5.3 we demonstrate the high accuracy of our model by comparing the histograms and the autocorrelation functions (acf) of frame size sequences corresponding to real and synthetic multiview data generated by the model. Finally, we also show that the proposed model closely matches the behaviour of an actual multiview source by examining their respective frame loss rate in network buffer constrained MVC streaming applications.

5.2 MVC Source Modeling

5.2.1 Description of the video source

An MVC source is composed of different video sequences captured simultaneously by multiple cameras. We denote the number of views as N_{View}. At each time instant, the MVC video sequence is composed of N_{View} pictures that may be transmitted together or just a subset of them, depending on the specific multiview application and the end user's needs. The video is organized into Groups Of Pictures (GOP)s, the length of which is denoted N_{GOP}, so that each GOP comprises $N_f \overset{\text{def}}{=} N_{\text{View}} \cdot N_{\text{GOP}}$ pictures. We use the terms frame and picture synonymously. The video is encoded using a fixed quantization step size without any rate control mechanism so that the resulting bitstream is completely VBR and its average bit-rate depends on the scene activity of the content. Our model characterizes the frame size sequence generated by the video source for all views.

5.2.2 Poisson-Hidden Markov Model (P-HMM)

Due to the fixed quantization step size and the scene variability that is typical of video content, the compressed multiview sequence does not represent a stationary stochastic process. In order to account for this, we model the video scene activity by means of a non-stationary HMM, in which the different states correspond to different levels of video scene activity. A random vector $x[n]$ is emitted in each state, where $x[n] \overset{\mathrm{def}}{=} [x_0[n], \ldots, x_{N_f-1}[n]]$ represents the set of frames in the n-th GOP of the compressed multiview content. The non-stationarity of the state sequence is achieved by modeling its state duration time via a probability mass function (*pmf*) different from the geometric distribution that in turn is typical for stationary conventional HMMs. Our choice is supported by the study in (26), where it is noted that the duration of video activity is not well described with a geometric distribution. In our case, we employ a Poisson distribution to model the state occupancy in order to maintain the same number of parameters (hence complexity) as for conventional HMMs.

Now, let us denote the number of states (*i.e.* different video activity levels) in our model as N_s and the state transition matrix as Π, where π_{ij} denotes the probability of transition from state i to state j. We impose $\pi_{ii} = 0$ since in our case the self-transition probabilities are governed by a (different) Poisson distribution and are denoted by $d_i[k] \overset{\mathrm{def}}{=} \frac{e^{-\lambda_i} \lambda_i^k}{k!}$. Given the current state of the model, say i, a random vector $x[n]$ is generated according to the *pmf* $b_i[x[n]]$. The mass functions $b_i[\cdot]$, $i = 1, \ldots, N_s$, have varying number of bins depending on the specific video frame to be generated (I, P, or B) in order to account better for their different complexities. Finally, π_i denotes the probability of the model being in state i.

The generation of synthetic content according to our model is summarized with the following steps:

1. A state is chosen according to the probability distribution $\pi_1, \pi_2, \ldots, \pi_{N_s}$. Assume state i is selected.

2. A state duration time, say k, is generated by the Poisson distribution conditioned on the current state, i.e., $d_i[k]$

3. The HMM stays in the state i for k time instances

4. k video frame vectors $x[n]$ are generated according to $b_i[\cdot]$

5. A state transition is performed according to Π

6. Go back to step 2

5.2.3 Parameter Estimation

The stage of parameter estimation is crucial in order to have a model able to describe an actual video source. Since the model is part of the HMM family, we can resort to one of the estimation algorithms employed for such models. In particular, an estimation procedure called Expectation-Maximization (EM) algorithm (18) is widely used for HMMs in order to find the maximum likelihood estimate of the parameter set.

In (49), a version of the EM algorithm for P-HMMs is introduced. Unfortunately, for long data sequences, as in video content, this specific algorithm becomes numerically unstable. For details, see (20; 46). Therefore, we derive a different EM algorithm for parameter estimation that does not exhibit numerical instability. Our algorithm is in major inspired by the work in (20) and represent its extension to the case of non-stationary hidden state duration. A brief summary of the proposed EM algorithm is presented in the following.

Suppose that we observe a video sequence composed of N GOPs. Let $x_0^{N-1} \overset{\text{def}}{=} \{x[n]\}_{n=0}^{N-1}$ denote the observed video traffic and $\Theta \in \boldsymbol{\Theta}$ the parameter set of our model, where $\boldsymbol{\Theta}$ is the parameter space and $\Theta \overset{\text{def}}{=} \{\Pi, \lambda_1, b_1[x], \pi_1, \ldots, \lambda_{N_s}, b_{N_s}[x], \pi_{N_s}\}$. The EM algorithm comprises two computational steps. The first one is an expectation step that computes the auxiliary likelihood function $Q(\Theta|\Theta^{(m)}) = E\{\log(\text{Prob}\{S, x, \Theta\})|x, \Theta^{(m)}\}$, where $S \in \mathbf{S}$ represents a plausible state sequence and $\Theta^{(m)}$ is the current (m-th) estimate of the parameter set. Then, a maximization step follows that maximizes the likelihood function, i.e.,

$$\Theta^{(m+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(m)}). \tag{5.1}$$

The algorithm iterates between the two steps until convergence of the parameter set $\Theta^{(m)}$ is achieved.

The specific computational steps of our EM algorithm, as applied to P-HMMs, comprise

1. The following forward probabilities are defined:[1]

$$\begin{cases} \alpha_n(i,k) \overset{\text{def}}{=} P(s_n = i, \ldots, s_{n+k} = i, s_{n+k+1} \neq i | x_0^n, \Theta^{(m)}) \\ \alpha_n(i) \overset{\text{def}}{=} P(s_n = i | x_0^n, \Theta^{(m)}). \end{cases}$$

   These quantities are calculated for $n = 0, \ldots, N-1$ and $k = 0, \ldots, N-n-1$ by a recursive algorithm that is not included here due to space constraints. This algorithm is similar to the one described in (20).

2. The a-posteriori probabilities:

$$\begin{cases} \gamma_n(i,k) & \overset{\text{def}}{=} P(s_n = i, \\ & \ldots, s_{n+k} = i, s_{n+k+1} \neq i | x_0^{N-1}, \Theta^{(m)}) \\ \xi_n(i,j,k) & \overset{\text{def}}{=} P(s_{n-1} = i,\ \ s_n = j, \\ & \ldots, s_{n+k} = j, s_{n+k+1} \neq j | x_0^{N-1}, \Theta^{(m)}) \end{cases} \tag{5.2}$$

   are calculated through a backward iteration similar to the one described in (20).

3. Finally, the parameter set $\Theta^{(m+1)}$ is calculated:

$$\pi_i = \sum_{k=0}^{N-1} \gamma_0(i,k) \tag{5.3}$$

$$\pi_{ij} = \frac{\sum_{n=1}^{N-1} \sum_{k=0}^{N-n-1} \xi_n(i,j,k)}{\sum_{\substack{j=1 \\ j \neq i}}^{N_s} \sum_{n=1}^{N-1} \sum_{k=0}^{N-n-1} \xi_n(i,j,k)} \tag{5.4}$$

$$b_i[x] = \frac{\sum_{n=0}^{N-1} \sum_{k=0}^{N-n-1} \gamma_n(i,k) \delta_x^{x[n]}}{\sum_{n=0}^{N-1} \sum_{k=0}^{N-n-1} \gamma_n(i,k)} \tag{5.5}$$

$$\lambda_i = \frac{\sum_{n=1}^{N-1} \sum_{k=0}^{N-n-2} \sum_{\substack{j=1 \\ j \neq i}}^{N_s} k\ \xi_n(j,i,k) + \sum_{k=0}^{N-1} k\ \gamma_0(i,k)}{\sum_{n=1}^{N-1} \sum_{k=0}^{N-n-2} \sum_{\substack{j=1 \\ j \neq i}}^{N_s} \xi_n(j,i,k) + \sum_{k=0}^{N-1} \gamma_0(i,k)}, \tag{5.6}$$

where $\delta_x^{x[n]}$ denotes the delta function.

[1] Our definitions differ from (49) in order to avoid numerical instability.

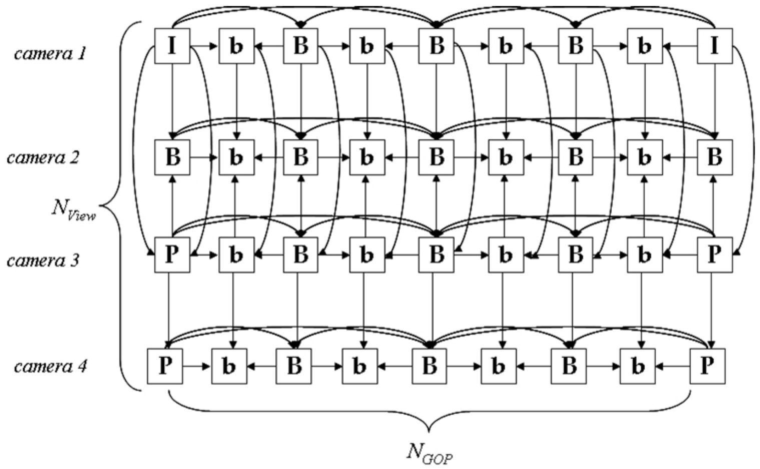

Figure 5.1: GOP and encoding structure. The arrows indicate the dependencies between frames.

5.3 Model Assessment

In this section, we examine the performance of our model. The multiview content employed in our experiments represents a concatenation of 5 test sequences (Akko & Kayo, Uli, Ballet, Breakdance, Pantomime) that exhibit different motion characteristics so that the concatenated content exhibits varying levels of video scene activity. The concatenated sequence is encoded using the reference encoder JMVC v.7.0 (30) at three different quantization levels $Q_s = 10, 20, 40$ in order to have three encoded sequences at respectively high, medium, and low quality. We have used the following encoding parameters: $N_{\text{GOP}} = 8$, $N_{\text{View}} = 4$, $N = 1016$. The GOP encoding structure is shown in Fig.5.1. The assessment is performed by comparing the real sequence to a synthetic one generated by the P-HMM when its parameters are estimated from the actual concatenated video content.

Because of the iterative nature of the EM algorithm employed for estimating the parameter set of our P-HMM, we need an initial solution, i.e., an initial estimate $\Theta^{(0)}$. This quantity is crucial for the proper convergence of the EM algorithm, as otherwise we may end up in a local maximum (20). We perform the initial estimation stage in two steps. First, we estimate the most likely state sequence of the P-HMM by assigning each GOP of the compressed content to one of the states according to the GOP's average bit-rate. Then, we estimate the P-HMM parameter set by time-averaging the multiview data associated with each state according to the previously estimated state sequence.

5.3.1 Model validation

We have performed the assessment of our model by comparing the actual multiview sequence and a synthetic one generated according to the model. We compare the two sequences by evaluating the histograms and the autocorrelation functions (acf) of their frame size values. If the model is able to mimic the statistical characteristics of the video source we expect to see the histogram and the acf of the real and the synthetic sequences to be very similar. We expect to observe a larger degree of similarity in the case of the acf, since the acf corresponds to an averaging operation performed over the different histogram bins. Due to space limitations, we have expressed the degree of statistical similarity by means of a percentage error both for the acf and the histogram values. By way of an example, the percentage error for the acf is calculated by the following expression:

$$\textit{percentage error} \overset{\text{def}}{=} \frac{\sum_k (\rho_r[k] - \rho_s[k])^2}{\sum_k \rho_r[k]^2} \cdot 100\%,$$

where $\rho_r[k]$ and $\rho_s[k]$ are respectively the acf for the real sequence and the synthetic one. A similar expression is used for calculating the percentage error between the histograms.

The percentage errors for the three sequences are shown in Tab.5.1. We have calculated the percentage errors by considering both the individual views as well as all views together. First, we would like to remark that most of the percentage errors are under 1%, which means a high degree of similarity between the actual and the synthetic data is achieved. Only for the high quality sequence histograms we have observed a slightly smaller degree of similarity. This is due to the fact that we still employ the same number of histogram bins, as in the cases of low and medium qualities, to sample the frame size values whose dynamic range has increased now due to the finer quantization. In essence, the synthetic data provides a coarser approximation of the frame sizes in this case. Furthermore, as seen from Tab.5.1 the acf percentage errors are generally lower than the corresponding histogram percentage errors, as expected and explained earlier. In summary, we

	$\mathbf{Q_s = 40}$		$\mathbf{Q_s = 20}$		$\mathbf{Q_s = 10}$	
	Hist	ACF	Hist	ACF	Hist	ACF
View 1	0.6%	0.4%	1%	0.06%	17%	0.5%
View 2	1%	3%	2%	0.03%	8%	0.4%
View 3	3%	3%	1%	0.02%	6%	0.2%
View 4	2%	3%	1%	0.02%	2%	0.2%
All Views	0.5%	0.5%	0.8%	0.03%	2%	0.5%

Table 5.1: Percentage errors for the acf and histograms of frame size sequences according to real and synthetic data.

can conclude that the proposed P-HMM model is able to accurately represent the statistical characteristics of the actual video source.

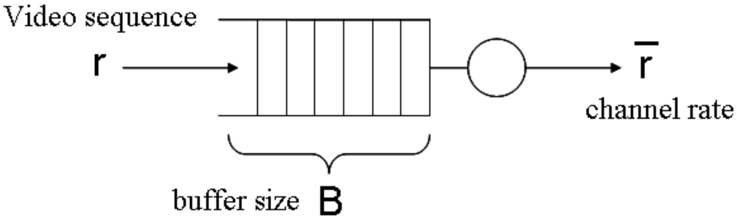

Figure 5.2: Buffer filling and depleting.

5.3.2 Buffer size dimensioning

In this section, we demonstrate that our model is able to reproduce the behaviour of actual video content in the context of streaming. Suppose that the video source is the input of a First In First Out (FIFO) network buffer of finite size B emptied at a constant rate $\bar{r}$, as shown in Fig.5.2.

One of the resources that should be determined in the stage of network dimensioning, in order to have the desired performance, is the appropriate size of this buffer (67)-(69). We show now that using the real sequence is equivalent to using the synthetic sequence for determining B. The buffer is fed with real or synthetic data and read out at a constant bit-rate equal to the average bit-rate of the real sequence. Then we compare the two sequences by means of the frame loss rate (an incoming frame is dropped when it is too large to be placed into the

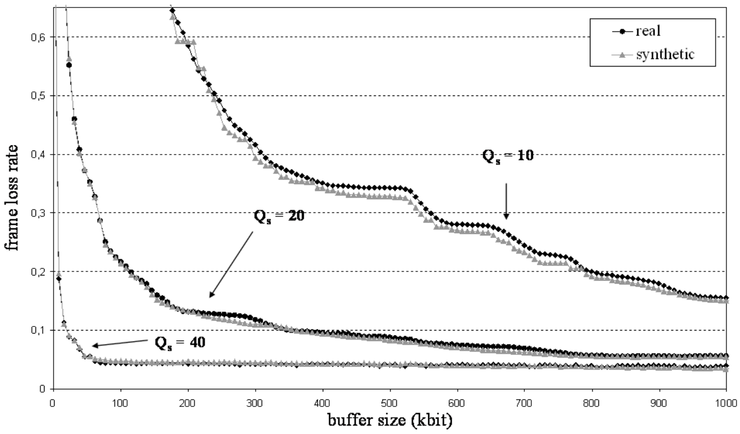

Figure 5.3: Frame loss rate for the synthetic sequence (gray - triangle) and the real sequence (black - circle), when all the views are transmitted.

buffer) as a function of the buffer size. We perform the test in two different cases: in the first case all the views are transmitted together (typical of a 3DTV-like application), in the second case the user watches a single view but he/she is able to switch among the views at his/her will. In order to have a fair comparison, we suppose that the view switching sequence that indicates the user's requests for view switching, is the same for both sequences (real or synthetic). Specifically, the viewing trajectory starts from view 1 and then switches to view 3, and then to view 2, and finally to view 1.

We remark that in the view-switching case, although the user watches only a single view at a time, he still receives some (or all) frames of the other views because they are needed for decoding the desired view. For this reason, network dimensioning is more difficult in this case and therefore having an accurate source model can be extremely useful. Fig.5.3 shows the results for the first case and Fig.5.4 for the second case. In both cases, the synthetic sequences have nearly the same frame loss rate as the real sequences. At high quality, we see a more stepwise shape of the frame loss rate for the synthetic sequence because of the smaller number of active bins of the $pmfb_i[\cdot]$. Still, a close resemblance to the frame loss rate of the real sequence is again observed.

Finally, we also examined our model for the network scenario where the buffer size is fixed and equals in size to 1000 ATM cells, while the output bit-rate is varying. Due to space constraints, we only show the results for the view-switching scenario in Fig.5.5. It can be seen that again the synthetic sequence's frame loss rate matches closely that of the actual video content. Moreover, the bin distribution of the *pmf* $b_i[\cdot]$ has a smaller influence in this network setup, as seen by the very close performances of the syntetic and real sequences in Fig.5.5 for the high quality case ($Q_s = 10$).

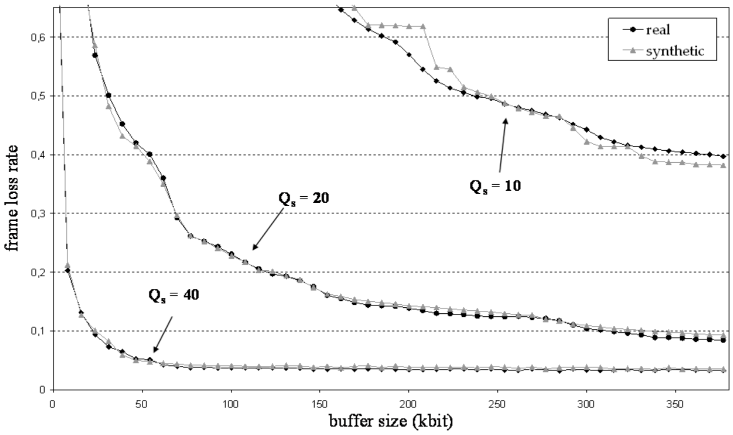

Figure 5.4: Frame loss rate for the synthetic sequence (gray - triangle) and the real sequence (black - circle) in the interactive TV case.

5.4 Conclusions

Our work provides the first traffic model of MVC compressed content. To this end, we have designed a non-stationary HMM in which each state corresponds to a different level of video scene activity and the state duration times are modeled with a Poisson distribution. We have derived, for the first time, a numerically stable version of the EM algorithm for estimating the parameters of a non-stationary HMM. Our modeling framework accurately captures the statistical properties described by histograms and the autocorrelation function of frame sizes in actual MVC content, both for each of the individual views as well as across all the views together. Furthermore, we have demonstrated that the proposed model closely matches the behavior of a real multiview source in buffer-constrained MVC streaming applications.

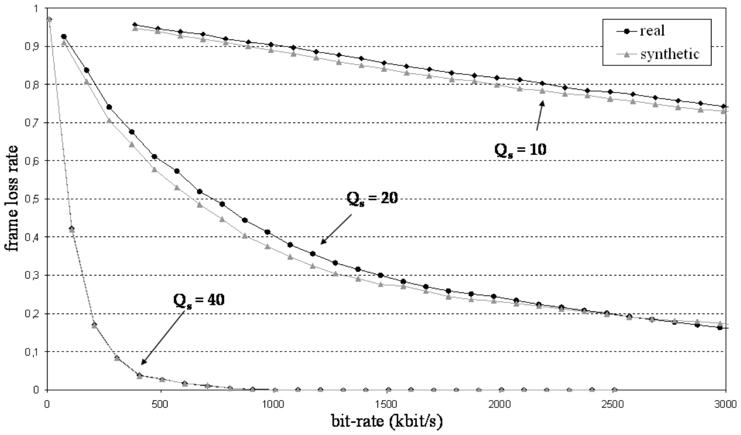

Figure 5.5: Frame loss rate for the synthetic sequence (gray - triangle) and the real sequence (black - circle) in the interactive TV case; fixed buffer size case.

Chapter 6

Conclusion

In this thesis we have analysed the main aspects of a mobile video streaming service, investigating the main issues and presenting a contribution for each one. After an introduction to the topics of the work, we have analysed the issues regarding the rate control in video streaming services. We have proposed a fair,from the point of view of the game theory, strategy to assign the coding mode and the number of the bits to the frames so as to minimize the distortion constrained to have a minimal required quality for each frame. Next, we have analyzed a novel error resilience scheme employing a multiple description coding based on polyphase subsampling and a robust edge-directed interpolation at the post-processing scheme. The interpolation takes into account the possibly errors due to channel transmission and defines a metric measuring the reliability. Simulations show that this approach improves video quality, both in terms of PSNR and perceptive quality. The other topic concerns a video traffic model for a video streaming source that dinamically switches among bitstreams at different bit-rates. The video model consists in a member of the family of Hidden Markov Models, in which the (hidden) state sequence describes the sequence of kind of GOPs, whereas state dependant multivariate Gaussian process models the frame size distribution. Model is assessed by comparing the pdfs and other averages to the respective of an actual observed source. Last topic of the contribution regards the novel video coding standard H.264/MVC, in particular a non stationary model of the video source is presented. We explicity model the duration time of the video activity, so as to obtain a non stationary model able to describe the video source. The Poisson distribution was chosen for the task, both for its simplicity and for the good results obtained. A pmf is introduced to describe the frame size distribution. Model validation shows that the model is able to reproduce video source behavior in a leaky buffer context.

Chapter 7

Appendices

A Frame layer statistics HMP

The compact model in (4.2), referring to the GOP layer, enables straightforward analytical evaluation of several statistical descriptions of video data. In this Appendix we summarize a few results which are of interest while describing the statistical behavior of a video source. As a first example, we report here the marginal frame size pdf. Let us denote by $x[m]$ the random variable representing the size in bits of the m-th frame of a video sequence obeying to the HMP generation mode reported in (4.2). The pdf $p(x[m])$ resulting in accordance to (4.2) turns out to be a mixture of normal distributions $\mathcal{N}\left(\|\mathbf{c}_\lambda\|_i, \|\Sigma_\lambda\|_{i,i}^2\right)$, each extracted with probability p_λ/N_{GOP}, that is:

$$p(x[m]) = \frac{1}{N_{\text{GOP}}}\sum_{\lambda} p_\lambda \sum_{i=0}^{N_{\text{GOP}}-1} \frac{1}{\sqrt{2\pi}\,\|\Sigma_\lambda\|_{i,i}} \exp\left(-\frac{(x[m]-\|\mathbf{c}_\lambda\|_i)^2}{2\,\|\Sigma_\lambda\|_{i,i}^2}\right) \tag{A.1}$$

In turn, according to (4.2), we have derived the bi-dimensional pdf $p(x[m], x[m+k])$; the analytical expressions of $p(x[m], x[m+k])$ for $k = 1$ and $k = N_{\text{GOP}}$ are reported in (A.2)-(A.3).

$$
\begin{aligned}
p(x[m], x[m+1]) = \frac{1}{N_{\mathrm{GOP}}} \sum_{\lambda=1}^{N_s} p_\lambda \\
\cdot \Big[\sum_{i=0}^{N_{\mathrm{GOP}}-2} \mathcal{N}\left(x[m], \|\mathbf{c}_\lambda\|_i, \|\Sigma_\lambda\|_{ii}^2\right) \cdot \mathcal{N}\left(x[m+1], \|\mathbf{c}_\lambda\|_{i+1}, \|\Sigma_\lambda\|_{i+1,i+1}^2\right) \\
+ \sum_{\mu=1}^{N_s} \pi_{\lambda\mu}\, \mathcal{N}\left(x[m], \|\mathbf{c}_\lambda\|_{N_{\mathrm{GOP}}-1}, \|\Sigma_\lambda\|_{N_{\mathrm{GOP}}-1,N_{\mathrm{GOP}}-1}^2\right) \cdot \mathcal{N}\left(x[m+1], \|\mathbf{c}_\mu\|_0, \|\Sigma_\mu\|_{0,0}^2\right) \Big]
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
p(x[m], x[m+N_{\mathrm{GOP}}]) = \frac{1}{N_{\mathrm{GOP}}} \sum_{\lambda=1}^{N_s} p_\lambda \sum_{\mu=1}^{N_s} \pi_{\lambda\mu} \\
\cdot \sum_{\imath=0}^{N_{\mathrm{GOP}}-1} \mathcal{N}\left(x[m], \|\mathbf{c}_\lambda\|_i, \|\Sigma_\lambda\|_{ii}^2\right) \mathcal{N}\left(x[m+1], \|\mathbf{c}_\mu\|_i, \|\Sigma_\mu\|_{i,i}^2\right)
\end{aligned}
\tag{A.3}
$$

Finally, we have considered the normalized autocorrelation:

$$
\rho_x[k] \overset{\mathrm{def}}{=} \frac{\displaystyle\sum_{m=0}^{M-k-1} \left(x[m] - \mathrm{E}\left\{x[m]\right\}\right)\left(x[m+k] - \mathrm{E}\left\{x[m]\right\}\right)}{\displaystyle\sum_{m=0}^{M-1} \left(x[m] - \mathrm{E}\left\{x[m]\right\}\right)^2}
\tag{A.4}
$$

and we have evaluated its asymptotical (large N) expected value, reported in (A.5) where we have denoted $\kappa \overset{\mathrm{def}}{=} (k+j)_{\mathrm{mod}\, N_{\mathrm{GOP}}}$.

$$
\begin{aligned}
E\{\rho_x[k]\} \simeq \Bigg[& \sum_{j=0}^{N_{\mathrm{GOP}}-1} \sum_{n=0}^{\left\lfloor \frac{M-k-1}{N_{\mathrm{GOP}}} \right\rfloor -1} \Bigg(\left\| R_{\mathbf{x}} \left[\left\lfloor \frac{k+j}{N_{\mathrm{GOP}}} \right\rfloor \right] \right\|_{\kappa,j} \\
& - \frac{1}{M} \sum_{p=0}^{N_{\mathrm{GOP}}-1} \sum_{l=0}^{M/N_{\mathrm{GOP}}-1} \Bigg(\left\| R_{\mathbf{x}} \left[n-l \right] \right\|_{j,p} + \left\| R_{\mathbf{x}} \left[n + \left\lfloor \frac{k+j}{N_{\mathrm{GOP}}} \right\rfloor - l \right] \right\|_{\kappa,p} \Bigg) \Bigg) \\
& + \sum_{j=0}^{(M-k-1) \bmod N_{\mathrm{GOP}}} \Bigg(\left\| R_{\mathbf{x}} \left[\left\lfloor \frac{k+j}{N_{\mathrm{GOP}}} \right\rfloor \right] \right\|_{\kappa,j} \\
& - \frac{1}{M} \sum_{p=0}^{N_{\mathrm{GOP}}-1} \sum_{l=0}^{M/N_{\mathrm{GOP}}-1} \Bigg(\left\| R_{\mathbf{x}} \left[\left\lfloor \frac{M-k-1}{N_{\mathrm{GOP}}} \right\rfloor - l \right] \right\|_{j,p} + \\
& \qquad \left\| R_{\mathbf{x}} \left[\left\lfloor \frac{M-k-1}{N_{\mathrm{GOP}}} \right\rfloor + \left\lfloor \frac{k+j}{N_{\mathrm{GOP}}} \right\rfloor - l \right] \right\|_{\kappa,p} \Bigg) \Bigg) \\
& + \frac{M-k}{M^2} \sum_{j=0}^{N_{\mathrm{GOP}}-1} \sum_{n=0}^{M/N_{\mathrm{GOP}}-1} \sum_{p=0}^{N_{\mathrm{GOP}}-1} \sum_{l=0}^{M/N_{\mathrm{GOP}}-1} \left\| R_{\mathbf{x}} \left[n-l \right] \right\|_{j,p} \Bigg] \\
& \cdot \Bigg(\frac{M}{N_{\mathrm{GOP}}} \sum_{i=0}^{N_{\mathrm{GOP}}-1} \left\| R_{\mathbf{x}} \left[0 \right] \right\|_{i,i} - \frac{1}{M} \sum_{n=0}^{M/N_{\mathrm{GOP}}-1} \sum_{i=0}^{N_{\mathrm{GOP}}-1} \sum_{m=0}^{M/N_{\mathrm{GOP}}-1} \sum_{j=0}^{N_{\mathrm{GOP}}-1} \left\| R_{\mathbf{x}} \left[n-m \right] \right\|_{i,j} \Bigg)^{-1}
\end{aligned}
\tag{A.5}
$$

In the Section devoted to numerical simulations, we have shown comparisons of these statistics with the corresponding sample statistics evaluated on real video traffic sources, thus assessing the adequateness of the HMP model to capture a few statistical features of a real video traffic source.

References

[1] 3rd Generation Partnership Project Technical Specification 3GPP TS 26.244, “Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)”.

[2] I. Ahmad, J. Luo, “On Using Game Theory to Optimize the Rate Control in Video Coding,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 16, February 2006.

[3] N. Ansari, H. Liu, Y. Q. Shi and Zhao, H., “On modeling MPEG video traffics”, *IEEE Trans. on Broadcasting*, vol.48, no.4, pp.337-347, December 2002.

[4] A. Baraldi, L. Bruzzone and P. Blonda, “A multiscale expectation-maximization semisupervised classifier suitable for badly posed image classification”, *IEEE Trans. Image Processing*, vol.15, n. 8, Aug. 2006.

[5] R. Bernardini, M. Durigon, R. Rinaldo, L. Celetto, A. Vitali, Polyphase spatial subsampling multiple description coding of video streams with H264, *Proc. ICIP 2004*, Singapore, October 24-27. 2004, pp. 3213-3216.

[6] J. M. Bioucas-Dias, “Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors”, *IEEE Trans. Image Processing*, vol.15, n. 4, April 2006.

[7] J. Cai, L.F. Chang, K. Chawla, and X. Qiu, “Providing dif- ferentiated services in EGPRS through packet scheduling,” *Proceedings of the IEEE Global Telecommunications Conference(GLOBECOM’00)*, vol. 3, San Francisco, USA, November-December 2000.

[8] Y. Chen, *et al.*, “The Emerging MVC Standard for 3D Video Services,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 13 pages, 2009.

[9] J. Chiang, Z.J. Wang, M.J. McKeown, “A hidden Markov, multivariate autoregressive (HMM-mAR) network framework for analysis of surface EMG (sEMG) data”, *IEEE Transactions on Signal Processing*, vol.56, no.8, pp.4069-4081, Aug. 2008.

[10] P.A. Chou, Z. Miao, “Rate-Distortion Optimized Streaming of Packetized Media,” *IEEE Trans. on Multimedia*, Vol. 8, April 2006.

[11] S. Colonnese, G. Panci, S. Rinauro and G. Scarano, “Optimal video coding for bit-rate switching applications: a game-theoretic approach,” *Proc. of IEEE Int. Symp. on a World of Wireless Mobile and Multimedia Networks*, Helsinky, Finland, June 2007.

[12] S. Colonnese, G. Panci, S. Rinauro, G. Scarano, “Modeling of H.264 video sources performing bitstream switching”, *PCS-2007*, Lisbon, Portugal, November 7-9, 2007.

[13] S. Colonnese, G. Panci, S. Rinauro, G. Scarano, “Markov model of H.264 video sources performing bit-rate switching”, *ICIP-2008*, San Diego, USA, October 12-15, 2008.

[14] S. Colonnese, S. Rinauro, L. Rossi, G. Scarano, “Markov model of H.264 video traffic”, *ISIVC-2008*, Bilbao, Spain, July 9-11, 2008.

[15] S. Colonnese, S. Rinauro, L. Rossi, G. Scarano, “H.264 Video Traffic Modeling Via Hidden Markov Process”, *EUSIPCO-2009*, Glasgow, UK, August 24-28, 2009.

[16] N.D. Doulamis *et all.*,“Efficient modeling of VBR MPEG-1 coded video sources”, *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol.10, no.1, pp.93-112, February 2000.

[17] M. Dai, D. Louguinov,“Analysis and modeling of MPEG-4 and H.264 multi-layer video traffic”, *INFOCOM-2005*, Miami, Florida, U.S., March 13-17, 2005.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from Incomplete data via the EM algorithm", *J. Roy. Statist. Soc. B*, vol.39, no.1, pp.1-38, 1977.

[19] P. A. Devijver, "Baum's forward-backward algorithm revisited", *Pattern Recogn. Lett.*, vol.3, pp.369-373, 1985.

[20] Y. Ephraim and N. Merhav, "Hidden Markov processes",

[21] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, Multiple description video coding for scalable and robust transmission Over IP, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 3, pp.321-334, March 2005.

[22] C. Francq and M. Roussignol, "On white noises driven by hidden Markov chains", *J. Time Ser. Anal.*, vol.18, no.6, pp.553-578, 1997.

[23] V. Goyal, Multiple description coding: compression meets the network, *IEEE Sig. Proc. Mag.*, vol. 18, pp. 74-93, Sept. 2001.

[24] H.264/AVC Codec Software Archive [Online], "http://iphome.hhi.de/suehring/tml/"

[25] D. S. Hands, "A Basic Multimedia Quality Model", *IEEE Trans. on Multimedia*, vol. 6, No. 6, Dec. 2004, pp:806-816.

[26] D. P. Heyman, T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic", *IEEE/ACM Trans. on Networking*, vol. 4 no. 1, 1996.

[27] D. Heyman, A. Tabatabai, T.V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks", *IEEE Trans. on circuits and systems for video technology*, vol.2, no.1, pp. 49-59, March 1992.

[28] M.R. Izquierdo, D.S. Reeves, "A survey of statistical source models for variable-bit-rate compressed video", *Multimedia Systems* , vol.7, no.3, May 1999, pp.199-213, May 1999.

[29] Z. Jiang, Y. Ge, Y. Li, “Max-utility wireless resource management for best-effort traffic,” *IEEE Trans. on Wireless Communications*, vol. 4 N. 1, January 2005.

[30] Joint multiview coding (JMVC) v7.0, available via CVS at garcon.ient.rwth-aachen.de.

[31] M. Karczewicz, R. Kurceren, “The SP- and SI-Frames Design for H.264/AVC” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, June 2003.

[32] J. Kim, R.M. Mersereau, and Y. Altunbasak, Distributed video streaming using multiple description coding and unequal error protection, *IEEE Trans. on Image Processing*, vol. 4, no. 7, pp. 849-861, July 2005.

[33] H. Koumaras *et all.*, “A Markov modified model of H.264 VBR video traffic”, IST Mobile Summit 2006, Mykonos, Greece.

[34] H. Koumaras *et all.*, “Analysis of H.264 video encoded traffic”, International Network Conference 2005, Samos, Greece.

[35] V. Krishnamurthy and R. J. Elliott, “A filtered EM algorithm for joint hidden Markov model and sinusoidal parameter estimation”, *IEEE Trans. Signal Processing*, vol.43, pp.353-358, Jan. 1995.

[36] V. Krishnamurthy and J. B. Moore, “On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure”, *IEEE Trans. Signal Processing*, vol.41, pp.2557-2573, Aug. 1993.

[37] M. M. Krunz and H. Hughes, ”A traffic model of MPEG coded VBR streams”, *Proc. ACM Sigmetrics/Performance’ 95 Conference*, pp.47-55, May 1995.

[38] D.S. Lee, C.M. Chen, C.Y. Tang, Weighted Fair Queueing and Compensation Techniques for Wireless Packet Switched Networks,” *IEEE Trans. on Vehicular Technology*, Vol. 56 n.1, January 2007.

[39] D. Le Gall, “MPEG: a video compression standard for multimedia applications”, *Commun. ACM*, Vol. 34, No. 4, April 1991.

[40] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition”, *Bell Syst. Tech. J.*, vol.62, no.4, pp. 1035-1074, Apr. 1983

[41] M. Li, and T.Q. Nguyen, Markov random field Model-Based edge-directed image interpolation, *IEEE Trans. on Image Processing*, vol. 17, no. 7, pp. 1121-1128, July 2008.

[42] D. Liu, E. I. Sàra and W. Sun, “Nested auto-regressive processes for MPEG-encoded video traffic modeling”, *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol.11, no.2, pp.169-183, February 2001.

[43] S. Ma, W. Gao, D. Zhao, and Y. Lu, “A Study on the Quantization Scheme in H.264/AVC and Its Application to Rate Control” *Advances in Multimedia Information Processing*, Vol. 3333, 2005.

[44] A. Matrawy *et al.*,, “MPEG4 traffic modeling using the transform expand sample methodology,” *IEEE 4th International Workshop on Networked Appliances*, Gaithersburg, 2002.

[45] J. Nash, “Two-person cooperative games,” *Econometrica*, vol. 21, January 1953.

[46] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pages 257-286, 1989.

[47] G. Ramamurthy, B. Sengupta, “Modeling and analysis of a variable bit rate multiplexer”, *IEEE INFOCOM*, vol.2 pp.817-827, May 1992.

[48] RFC 3984, “RTP payload format for H.264 video ”, Wenger S. et al., February 2005.

[49] M. Russell,R. Moore, “Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition,” ICASSP 85, 1985.

[50] E. Setton, B. Girod, “Rate-Distortion Analysis and Streaming of SP and SI Frames,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 16, June 2006.

[51] H. R. Sheikh, A. C. Bovik, “Image information and visual quality”, *IEEE Trans. on Image Processing*, vol. 15, No.2, Feb. 2006, pp:430-444.

[52] A. Stefanescu and M.W. Stefanescu, “The arbitrated solution for multi-objective

[53] T. Stockhammer, G. Liebl,M. Walter, “Optimized H.264/AVC-based bit stream switching for mobile video streaming,” *EURASIP Journal on Applied Signal Processing*, Vol. 2006, January 2006. convex programming,”

[54] C. Su, J.J. Yao, and H.H. Chen, H.264/AVC-Based multiple description coding scheme, *Proc. ICIP 2007*, San Antonio, September 16-19. 2007, pp. 265-268.

[55] X. Sun, S. Li, F. Wu, J. Shen, W. Goo, “The Improved SP Frame Coding Technique for the JVT Standard” *Proc. of IEEE Int. Conf. on Image Processing*, Atlanta, GA, US, October 8-11, 2006. *Rev. Roum. Math. Pure Applicat.*, vol. 29, 1984.

[56] W. Tan, G. Cheung, “SP-Frame Selection for Video Streaming over Burst-loss Networks” *Proc. of IEEE Int. Symp. on Multimedia*, Irvine, California, US, December 12-14, 2005.

[57] W. Tan, B. Shen,, “Method to Improve Coding Efficiency of SP Frames” *Proc. of IEEE Int. Conf. on Image Processing*, Atlanta, GA, US, October 8-11, 2006.

[58] T. Tillo, M. Grangetto, and G. Olmo, Redundant slice optimal allocation for H.264 multiple description coding, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp.59-70, Jan. 2008.

[59] T. Tillo, and G. Olmo, Data-dependent pre- and post-processing multiple description coding of images, *IEEE Trans. on Image Processing*, vol. 16, no. 5, pp. 1269-1280, May 2007.

[60] A. Vitali, Multiple description coding: a new technology for video streaming over the Internet, *EBU Technical Review*, Bergamo, October 2007.

[61] Y. Wang, A.R. Reibman, and S. Lin, Multiple description coding for video delivery, *Proc. of the IEEE*, vol. 93, no. 1, pp. 57-70, Jan. 2005.

[62] S. Wenger, H.264/AVC over IP, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, pp. 645-656, July 2003.

[63] T. Wiegand *et all.*, "Overview of the H.264 video coding standard", *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol.13, no.7, pp.560-576, July 2003.

[64] T. Wiegand, B. Girod, "Lagrange multiplier Selection in Hybrid Video coder Control," *IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[65] T. Wiegand, G.J. Sullivan, and A. Luthra, "ITU-T recommendation and final draft international standard of joint video specification, Joint Video Team Doc. JVT-G050r1",*ITU-T Rec. H.264 ISO/IEC 14-496-10 AVC*, June 2003.

[66] N. A. Woods, N. P. Galatsanos and A. K. Katsaggelos, "Stochastic methods for joint registration, restoration, and interpolation of multiple undersampled images", *IEEE Trans. Image Processing*, vol.15, n. 1, Jan. 2006.

[67] S. Xu, Z. Huang, "A Gamma Autoregressive Video Model on ATM Networks," *IEEE Trans. on Circ. and Sys. for Video Tech.*, Vol. 8, No. 2, pp. 138-142, 1998.

[68] F. Yegenoglu, B. Jabbari, Y.Q. Zhang, "Motion-classified autoregressive modeling of variable bit-rate video", *IEEE Trans. Circuits Systems Video Technol.*, vol.3, n.1, pp.42-53, Febraury 1993.

[69] Z. Zang, *et al.*, "Smoothing, Statistical Multiplexing, and Call Admission Control for Stored Video," *IEEE Jour. on Select. Areas in Commun.*, vol. 15, no. 6, 19 pages, 1997.

List of Figures