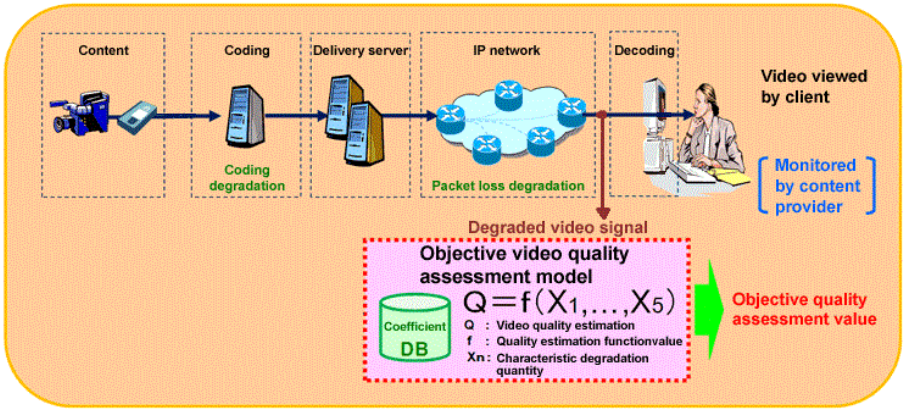
INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Video Quality Evaluation in IP networks

Miguel Filipe Chan Chin



Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri

Presidente: Prof. José Manuel Bioucas Dias

Orientador: Prof. Maria Paula dos Santos Queluz Rodrigues

Co-orientadores: Prof. Tomás Gomes da Silva Serpa Brandão
Prof. António José Castelo Branco Rodrigues

Vogal: Prof. Paulo Jorge Lourenço Nunes

Abril 2012

Acknowledgements

I would like to thank Professor Maria Paula Queluz and Professor Tomás Gomes Brandão for their supervision, availability and advising during the course of this Thesis. Thank you for all the suggestions, critics and support given. Without them this work would've not been possible.

I would also like to thank all my friends and colleagues, with whom I shared good and bad moments during the course of my academic life. I've learned a lot from these experiences and they will certainly help me in the future.

And finally, my family, for all the support given during my academic life and, the now starting, professional life.

To all of you, I thank you.

Miguel Chin

Abstract

Due to the growing demand for video applications and services, the need for accurate and effective video quality assessment metrics is increasing. This thesis proposes and evaluates no-reference objective quality metrics for encoded videos transmitted over a lossy channel. For this purpose, the metrics consider both the effects of H.264/AVC video encoding and packet losses over IP networks. All the developed algorithms predict the video perceived quality based on elements extracted from the bitstream and on the information about the packet losses, taken from the headers of the received packets.

Five quality assessment metrics for H.264/AVC videos impaired by packet losses are proposed and evaluated: 1) a simple model that accounts for the packet loss ratio; 2) a model that considers the frame type where the correspondent losses occur; 3) a model that considers the frame type and the movement in the video under analysis; 4) a model that considers the frame type, the movement in the video and the dependencies between frames; 5) a model that considers the frame type and statistical metrics taken from the packet loss pattern. The fifth model provided the best results, with quality predictions well correlated with subjective assessment data.

Keywords

No-reference quality assessment, H.264/AVC, Video over IP, Packet losses.

Resumo

Devido ao aumento da procura por aplicações e serviços de vídeo, a necessidade de métricas de avaliação de qualidade de vídeo tem vindo a crescer. Nesta dissertação propõem-se métricas objectivas de qualidade para vídeo codificado e transmitido num canal com perdas, sem recorrer a sinais de referência. Com esta finalidade, as métrica têm em conta as perdas resultantes da codificação H.264/AVC e as perdas de pacote numa rede IP. Os algoritmos envolvidos estimam a qualidade do vídeo baseando-se em elementos extraidos do fluxo binário do vídeo codificado e na informação retirada dos cabeçalhos dos pacotes recebidos.

São propostas e avaliadas cinco métricas de qualidade para vídeos H.264/AVC afectados por perdas de pacotes: 1) um modelo simples que tem em conta a taxa de perdas de pacote; 2) um modelo que considera o tipo da trama onde as perdas ocorrem; 3) um modelo que considera o tipo de trama e o movimento no vídeo em análise; 4) um modelo que considera o tipo de trama, o movimento e as dependências entre tramas; 5) um modelo que considera o tipo de trama e medidas estatísticas retiradas do padrão das perdas de pacotes. O quinto modelo produziu os melhores resultados, com predições de qualidade bem correlacionadas com os valores resultantes de avaliações subjectivas.

Palavras-chave

Avaliação de qualidade sem referência, H.264/AVC, Vídeo em redes IP, Perdas de pacote,

Table of Contents

List of Figures

List of Tables

List of Acronyms

BER	Bit Error Rate
B-Frame	Bidirectionally Predicted Frame
DCT	Discrete Cosine Transform
DSIS	Double Stimulus Impairment Scale
EPFL	Ecole Polytechnique Fédérale de Lausanne
FMO	Flexible Macroblock Ordering
FR	Full Reference
IETF	Internet Engineering Task Force
IP	Internet Protocol
I-Frame	Intra Frame
JVT	Joint Video Team
MB	Macroblock
MOS	Mean Opinion Score
MOSp	Predicted MOS
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
MTU	Maximum Transfer Unit
MV	Motion Vector
NAL	Network Abstraction Layer
NR	No Reference
OSI	Open System Interconnection
P-Frame	Predicted Frame
PL	Packet Loss
PLR	Packet Loss Ratio
PoliMi	Politecnico di Milano
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
RMS	Root Mean Square

RR	Reduced Reference
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SI	Spatial Information
SS	Single Stimulus
TCP	Transmission Control Protocol
TI	Temporal Information
UDP	User Datagram Protocol
VCL	Video Coding Layer
VQEG	Video Quality Experts Group

Chapter 1

Introduction

Video transmission over Internet Protocol (IP) networks is a growing market. This translates into an increasing number of service providers using the Internet Protocol Television (IPTV) system in order to allow services like live television or Video on Demand [Cisc08]. As competition between the service providers increases, meeting and exceeding the customers' expectations becomes more relevant; since the success of a service provider is strongly dependent on the entire end user experience, there is clearly a need for Quality of Experience (QoE) evaluation methods as they provide an indication of the customers' satisfaction. These QoE evaluation methods also allow the service providers to control the end-to-end perceptual video quality and to allocate the network resources according to the user satisfaction needs.

Since the end users are the target consumers of the service providers, they are naturally the most reliable source for quality assessment. However, gathering QoE data from users is not an easy task as it requires subjective quality assessment tests. These tests must be performed in controlled environments and require quality evaluation done by several users. In consequence, they can be very time consuming and expensive and, additionally, they cannot be performed in real-time. An alternative to subjective quality assessment is to automatically score the users perceived quality using objective metrics.

The user's QoE in visual communications is determined by a variety of factors. The channel zapping time, the service availability, the audio signal quality and its synchronization with the video signal, the set-top box boot time and its interface, all of them are important QoE indicators. Nevertheless, one of the key QoE factors is the quality of the pictures displayed on the screen.

Objective picture quality metrics can be categorized into full reference (FR), reduced reference (RR) and no-reference (NR). FR metrics require both the original and distorted video to compute the video quality. They are usually used for benchmarking video processing algorithms, such as lossy encoding, and media distribution networks during the testing phases. When the distribution network is setup and starts working, it's not appropriate to use FR metrics to evaluate the quality of the receive video. This is because the original video is usually unavailable at the receiving end of the network. RR metrics require only some information about the original video, while NR metrics rely only on the received video. RR and NR metrics are adequate for in-service quality monitoring at the user end and in-service network mid-point monitoring. NR metrics have the advantage of not requiring additional bandwidth, unlike the RR metrics. Thus, NR metrics can be considered as the most practical method for assessing network video quality.

Already developed metrics for video quality assessment are usually based on information extracted from the packet headers and/or from the video bitstream and/or from the decoded video. Reference [WiMo08] proposes a classification of quality metrics based on the type of used information, which is sketched in Figure 1.

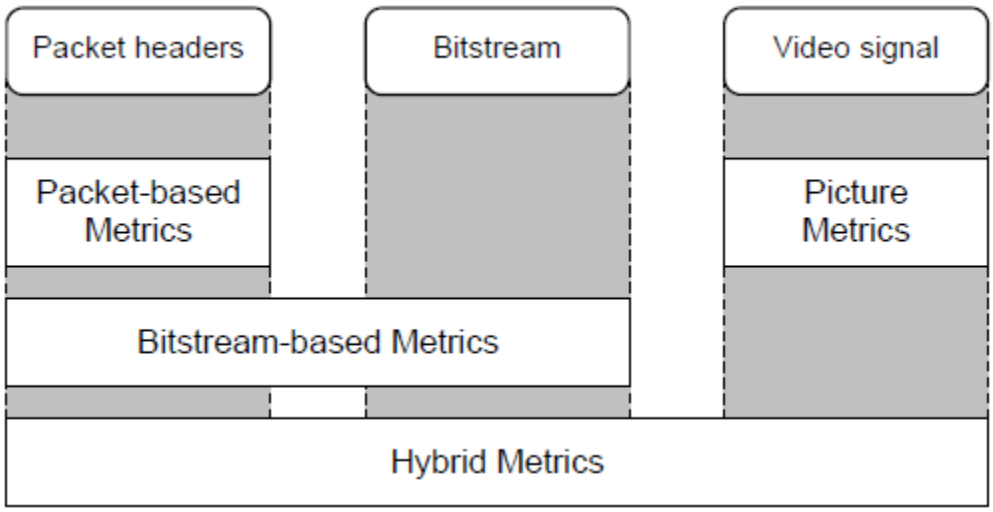


Figure 1 - Classification of quality metrics (extracted from [WiMo08])

The metrics that use information from the packet headers and bitstream usually need to be adjusted for specific encoders and transport protocols. In this thesis we intended to develop a bitstream-based NR quality metric for H.264/AVC encoded video when transmitted over an IP network. We started by studying the effects of lossy H.264/AVC video encoding, resulting in a NR quality metric that accounts for the compression impact on video quality. Afterwards, the effects of

packet losses over IP networks were studied and the conclusions obtained were used to improve the previously proposed NR quality metric, by considering both encoding and transmission losses. In this context, the dissertation is organized as follows.

Chapter 2 presents an overview of video quality assessment; the main reasons for video quality degradation in IP networks are described and methods, used by the decoder and H.264/AVC encoder to resist and conceal occurring errors, are outlined. Additionally, the chapter overviews already proposed NR metrics for transmitted video and a set of performance measures used to evaluate objective quality metrics.

In chapter 3, new NR metrics for encoded video, accounting only for compression distortions, are described, evaluated and compared. Two of them were originally proposed as FR metrics in ([BRK09] and [WP02]). However, they were modified in order to become NR metrics by using the error estimation module proposed in [BrQu10].

In chapter 4, new NR metrics for transmitted video over an IP network, and accounting for both compression and transmission distortions, are described, evaluated and compared. These metrics take into account various factors such as video content characteristics and packet loss statistics.

In chapter 5, the main conclusions of the thesis are given, and some proposals of future work are put forward.

The work developed in this thesis has been published in [BCQ11] and [CBQ12].

Chapter 2

Video Quality Overview

2.1 Introduction

The quality of a transmitted video is important not only to the end user, but also to the service provider. On the video producer's side the quality of a video is typically high; however, when the end user receives and sees the video, its quality is usually lower. This can happen for various reasons, but mainly it is due to compression and transmission losses (Figure 2). Video compression is necessary since a raw video generates a great amount of data making it unbearable for transmission or storage. Video compression reduces the amount of data necessary to represent the video by exploiting spatial, temporal and statistical redundancy as well as reducing the irrelevancy on the video signal. However, removing too much information may decrease the video quality.

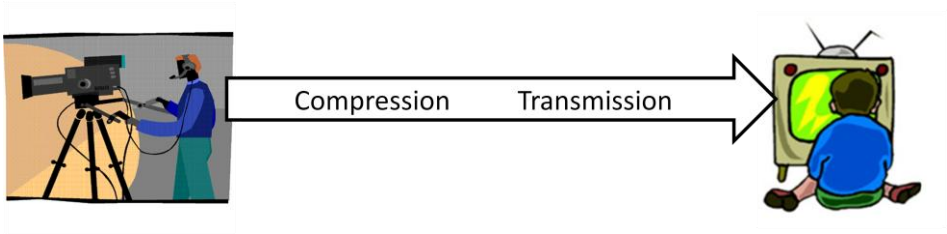


Figure 2 - The path from the video producer to the end user

After compression, the video can then be stored or transmitted. If transmitted, the quality of the video can still decrease due to transmission losses; these are mainly due to signal attenuation and distortion or, in the case of transmission over IP networks, due to lost packets. The quantification of perceived quality due to compression related factors, such as video codec type and video coding parameters, has been intensively studied and accurate metrics have been developed (e.g., [Roq09] and the references there included); however, there is still work to be done in what concerns the prediction of the perceived loss of quality due to transmission impairment factors.

This chapter is organized as follows. Section 2 gives an overview of the H.264/AVC standard, which is the codec used in this thesis for video compression. Section 3 analyzes the origin of transmission losses in IP networks. Section 4 describes the techniques that can be used by the H.264/AVC encoder, in order to prevent errors caused by the losses, and the techniques that can be used by the H.264/AVC decoder to conceal those errors. Section 5 reviews a few proposals of NR quality metrics accounting for transmission losses.

2.2 Overview of the H.264/AVC

H.264/AVC is a video compression standard developed by the ITU-T Video Coding Experts Group together with the ISO/IEC Moving Picture Experts Group (MPEG) in a partnership known as the Joint Video Team (JVT), formed in 2001 [WSBL03]. The objective of the JVT was to develop an advanced video coding specification capable of coding rectangular video with higher compression efficiency (about 50% less rate for the same quality regarding existing standards such as H.263, MPEG-2 video and MPEG-4 Visual). Another objective was to have good flexibility in terms of efficiency-complexity trade-offs in order to allow the standard to be applied on a wide variety of applications, such as:

- Broadcast over cable, satellite, terrestrial, etc.
- Storage on optical and magnetic devices, DVD, blue-ray, etc.
- Conversational services over ISDN, Ethernet, LAN, DSL, wireless and mobile networks, etc.
- Video-on-demand or multimedia streaming services over ISDN, DSL, LAN, wireless networks, etc.

To address the need for flexibility and customizability, the H.264/AVC design covers a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL). The VCL was designed in a way to represent the video content efficiently; the NAL formats the VCL representation of the video so it can be compatible with various transport protocols or storage media (Figure 3).

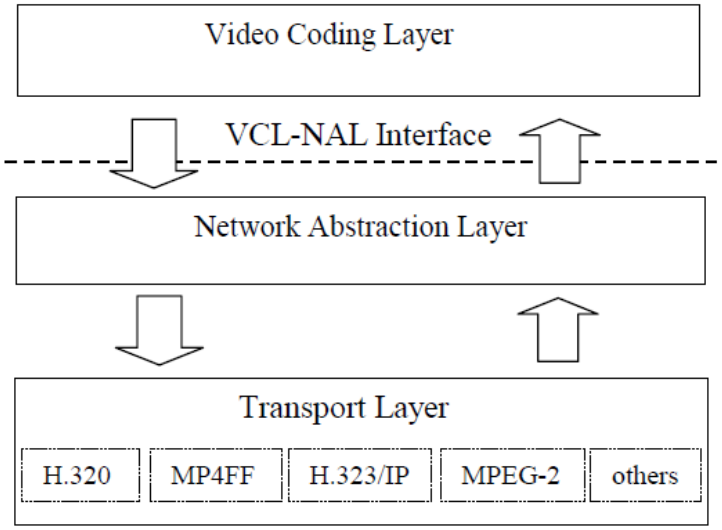


Figure 3 – The VCL and NAL layers

At the VCL, efficient video compression is achieved by exploiting the spatial and temporal redundancies of the video. Much like previous standards, H.264/AVC is based on a block-based coding approach. This means each frame of the video is represented by block shaped units called *macroblocks*, being each *macroblock* represented by 16x16 luminance pixels and by two 8x8 chrominance samples. The standard specifically defines three types of frames (Figure 4):

- **Intra frames** (I-Frames), exploit spatial redundancy and are coded using only information within the frame. These frames are also used for random access since they do not require information from previous frames. I-Frames provide the less compression among the three frame types.
- **Predicted Frames** (P-Frames), not only exploit spatial redundancy but also temporal redundancy. This is done by using information from previous I or P frames.
- **Bidirectionally Predicted Frames** (B-Frames), also exploit spatial and temporal redundancies but they may use information from past, as well as from future I or P frames. B-Frames provide the highest compression among the three frame types.

Figure 4 - I, P and B frames

The three frame types were also defined on previous standards, but H.264/AVC improved on them by adding some new features, such as: multiple reference frame motion compensation (Figure 5) and the ability to use a B-Frame as a reference frame.

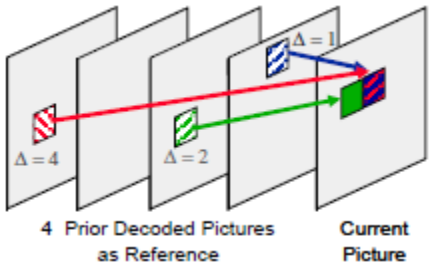


Figure 5 - Multiple reference frame motion compensation

As previously mentioned, the NAL adapts the compressed data from the VCL, so it can be compatible with various transport protocols. For video transmission over IP networks, the protocols are defined by the International Organization for Standardization (ISO) and by the Internet Engineering Task Force (IETF). Protocols were defined for three layers of the Open Systems Interconnection (OSI) model (Figure 6): Real-time Transport Protocol (RTP) on the application layer, User Datagram Protocol (UDP) on the transport layer and IP on the network layer [Weng03].

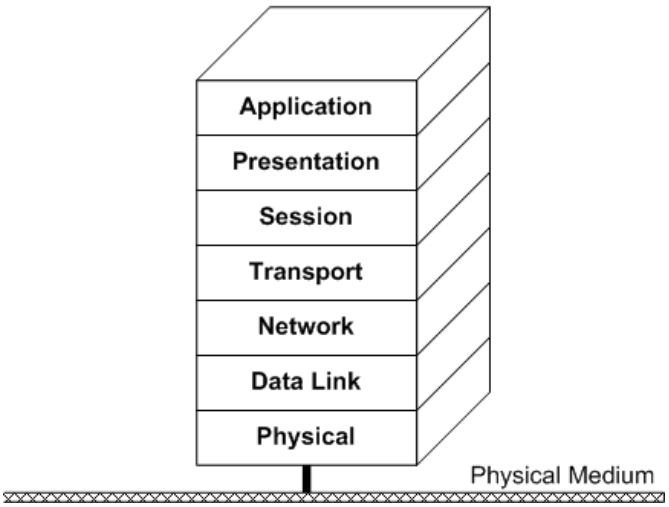


Figure 6 - The OSI reference model

The RTP is used in conjunction with the RTP Control Protocol (RTCP). RTP is designed for data transfer and the RTCP for control messages. For applications where timing is essential (e.g., video streaming) RTP is the protocol mostly used. This is because RTP allows the numbering of sent packets, which lets the order of received packets to be the same as the sent packets. RTP also allows the use of timestamps to be sure that a received packet is played when it is supposed to. Additionally, it enables the detection of lost packets. The RTCP is used to periodically send control information and QoS parameters such as, number of lost packets, inter-arrival jitter, delay, etc. With this information, the sender can optimize its transmission e.g., by adjusting the bit rate.

On the transport layer, IP networks commonly use the Transmission Control Protocol (TCP) or the UDP. TCP is byte stream oriented and guarantees delivery, which is achieved by retransmission and timeout mechanisms for error control. However, the protocol has an unpredictable delay making it unsuitable for media streaming. On the other hand, UDP is packet oriented and doesn't guarantee delivery since packets can be lost, duplicated or re-ordered. As a trade off, the delay of the delivered packets is more predictable and smaller when compared to TCP. Since delay is extremely important in media streaming, the higher layer protocol RTP is used over UDP to transmit the media data, while RTCP is usually used over TCP to send control messages.

The Internet Protocol IP is obviously used on IP networks. It enables packet delivery between hosts in the network through a set of routers using IP addressing which can be interpreted by everyone. Each packet contains, in addition to its data, a header, which includes its source and destination IP addresses. The size of each packet is limited by the Maximum Transfer Unit (MTU), which is the largest packet size that can be transmitted over the network and its value varies according to the type of protocol and network. However, if the data to be transmitted is bigger than the MTU, the protocol is responsible for splitting and recombining the data. The protocol offers a best effort routing, since the routers may discard packets, which are interpreted by the receiver as Packet Losses (PL). The reason behind these transmission losses is analyzed in section 2.3.

2.3 The origins of video losses

As previously mentioned, the decrease of the video quality when transmitted over an IP network can be due to compression losses and transmission losses. When a H.264/AVC encoder exploits the spatial redundancy it uses a type of coding based on the Discrete Cosine Transform (DCT). The coefficients resulting from the transform are quantized in order to remove irrelevancy. However, this quantization can reduce the quality of the video since some information is discarded resulting in compression losses, that can manifest as visible picture artefacts.

Concerning transmission losses in IP networks, packet losses occur mainly due to three factors [KGPL06]:

- Occasional bit errors caused by low noise margin or equipment failure.

- Buffer overflow or packet delay caused by congestion in the network.
- Rerouting to get around breakdowns or bottlenecks in the network.

Since the decoder on the receiving side needs that packets arrive in time to be displayed, packets too much delayed are discarded. In Figure 7 an example of this situation is shown: packet number 3 didn't arrive in time resulting in its drop.

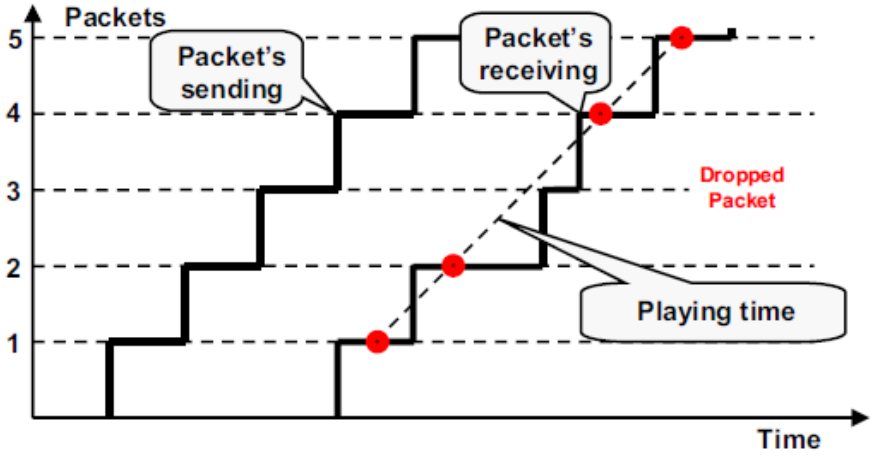


Figure 7 - Packet loss due to delay

2.4 Error resilience and concealment techniques

The H.264/AVC standard provides error resilience schemes [KXMP06] in an attempt to minimize the consequences of transmission losses. These are mainly contained in the VCL and some of them have been used in previous standards.

Some of the error resilience techniques are:

- **Semantics and syntax** – The standard defines all the syntax elements, such as the packet number, the timestamp, block coefficients and MV.
- **Intra-frame refreshing** – The use of I-Frames is a great tool to stop error propagation (Figure 8), since they are independently coded without temporal prediction and can reset the prediction process.

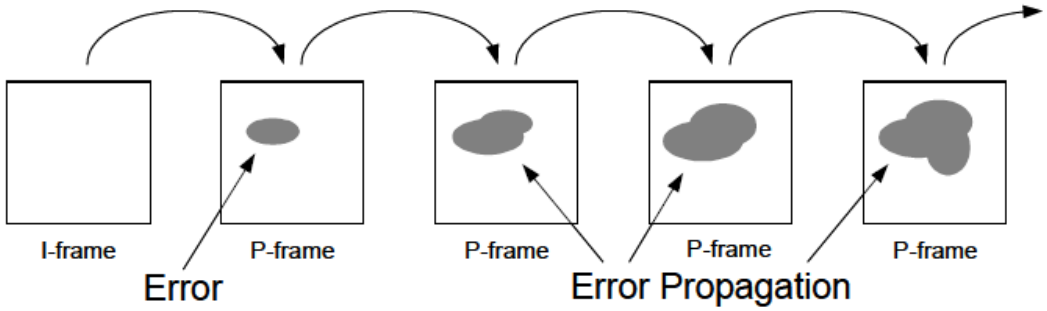


Figure 8 - Error propagation from a single error

- **Slice structuring** – The H.264/AVC encoder can organize MBs in groups called *slices* whose size is less or equal to the MTU.
- **Flexible *macroblock* ordering** (FMO) – Scattered errors are better concealed when compared with errors concentrated in a small region. FMO takes advantage of that fact by allowing a frame to be split into many MB scanning patterns such as interleaved slices and checker-board (Figure 9)

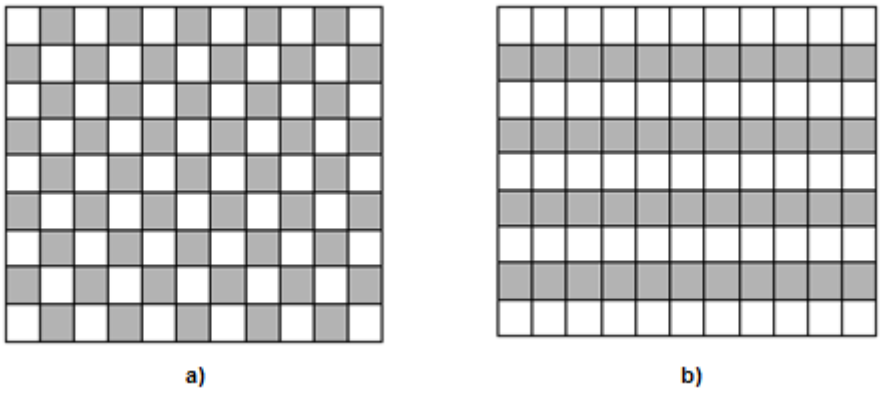


Figure 9 - FMO a) checker-board mode b) interleaving mode

When the decoder detects an error it can use an error concealment technique to try to make the error unnoticed. Some basic techniques are inter-frame prediction and intra-frame prediction. Inter-frame prediction uses information from previous frames, while intra-frame prediction uses information from the same frame in order to predict the content of lost MBs. Figure 10 - Intra-frame prediction depicts an example of error concealment using intra-frame prediction. The black blocks in Figure 10 b) represent lost MBs; the decoder used the information from the surrounding MBs to predict the content shown on Figure 10 c). By comparing with Figure 10 a), we can see that for this situation, intra-frame prediction wasn't enough since the decoder was unable to predict the human face on the video.

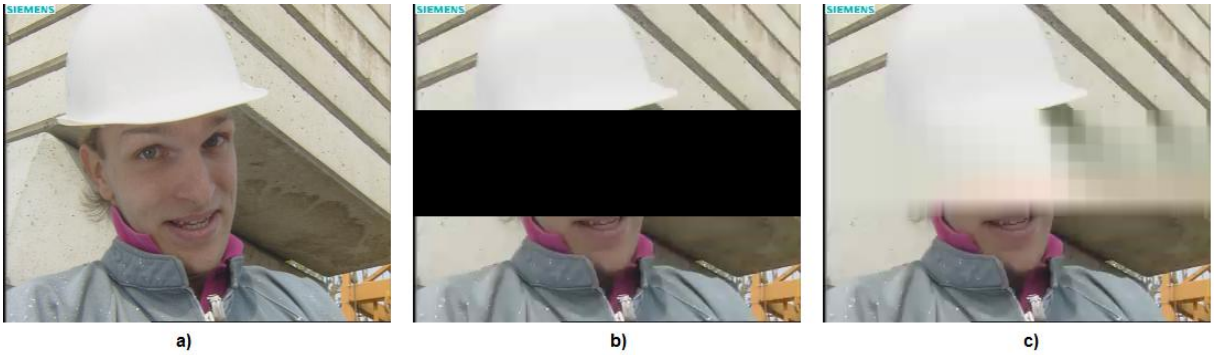


Figure 10 - Intra-frame prediction a) original b) losses detected c) prediction

Figure 11 depicts an example of error concealment using inter-frame prediction. Once again, the black blocks represent lost MBs, but this time the decoder used information from another frame to predict the content shown in Figure 11 c). By comparing with Figure 11 a), we can see that the decoder was able to do a rough prediction on the human face on the video.

Figure 11 - Inter-frame prediction a) original b) losses detected c) prediction

2.5 No-reference objective quality metrics

The effects of compression on video quality have been intensively studied and accurate metrics have been developed (e.g., [BrQu10][BrQu08] and the references included). On the other hand, the effects of transmission on video quality still need to be more investigated. There are simple metrics, such as the *bit error rate* (BER) or the *packet loss ratio* (PLR), which can be used to roughly predict the video quality. However, these metrics are not accurate since they do not take into account many factors that may affect the quality, such as:

- Burstiness of packet losses: distinct loss patterns may cause different perceptible video quality at the same PLR.
- Number of frames affected by packet losses.

- Type of frame associated to each lost packet.

At the present time, there are no standardized procedures for no-reference video quality assessment, although an intensive work on that subject is being performed by ITU-T and ITU-R through study/working groups SG9, SG12, and WP6C. The most recent standards from ITU, released in 2008 under the designations ITU-T Recommendations J.246 [ITUT08] and J.247 [ITU08] standardize a reduced reference and a set of full reference video quality assessment metrics, respectively. The closest standard that is related with NR image quality assessment is ITU-T Recommendation G.1070 [ITUT07], which presents a quality model for video telephony applications. It takes into account both audio and video parameters, as well as the delay between them. For video quality estimation, it requires parameters like the video's bit rate, frame rate and PLR. Additionally, the estimation function requires adjustments before it can be used. These adjustments are dependent on codec type, video format, key frame interval and video display size.

The G.1070 algorithm has spawned a few variations such as the ones proposed in [JoAr10], [BeMo10] and [YoZX09]. In [YoZX09] the burstiness of packet losses factor is exploited on the proposed quality estimation model.

Concerning scientific publications, three NR methods were proposed in [EVS04] to estimate mean squared error due to packet losses, directly from the video bitstream; the first method uses only network-level measurements (like PLR); the second method extracts the spatio–temporal extent of the impact of the loss; the third one extracts sequence-specific information including spatio–temporal activity and the effects of error propagation. Winkler and Mohandas proposed in [WiMo08] a no-reference metric – the *V–factor* –oriented to packetized transmission of MPEG-2 and H.264/AVC video. The metric uses information collected from the packet headers, from the bitstream and from the decoded video, and combines the collected data in order to obtain a quality score. However, since the metric was developed for commercial purposes, there are not many details on its implementation. More recently, in [YWXW10], a quality measure for networked video is introduced using information extracted from the compressed bit stream without resorting to complete video decoding. It accounts for three key factors which affect the overall perceived picture quality of networked video, namely, picture distortion caused by quantization, quality degradation due to packet loss and error propagation, and temporal effects of the human visual system.

2.6 Performance metrics

The quality of video quality prediction models depends on how well the predicted MOS correlate with the subjective test results. To quantify the performance of each model, the performance metrics proposed by the video quality experts group (VQEG) in [VQEG03] are typically used. They evaluate a model's prediction accuracy, prediction monotonicity, prediction consistency and root mean square error (RMS).

The Pearson correlation coefficient is a measure of the linear correlation between two variables. If the variables have no linear correlation between them, then the Pearson coefficient is 0; if the variables have a perfect linear correlation, then the Pearson coefficient is 1. However, achieving a perfect linear correlation between the predicted MOS and the subjective MOS is very difficult. A Pearson coefficient between 0.9 and 1 is usually considered as acceptable; any MOS prediction model scoring a Pearson coefficient outside this range is typically considered to be an inadequate model. The Pearson coefficient is given by:

$$P_c = \frac{\left(\sum_{i=1}^{N}(x_i \times y_i)\right) - \left(\frac{1}{N} \times \sum_{i=1}^{N} x_i \times \sum_{i=1}^{N} y_i\right)}{\sqrt{\left[\left(\sum_{i=1}^{N} {x_i}^2\right) - \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)^2\right] \times \left[\left(\sum_{i=1}^{N} {y_i}^2\right) - \frac{1}{N}\left(\sum_{i=1}^{N} y_i\right)^2\right]}} \quad (1)$$

where x_i is the subjective MOS, y_i is the predicted MOS and N is the total number of video sequences evaluated.

The Spearman correlation coefficient is a measure of how well the relation between two variables, in this case the subjective MOS and the predicted MOS, can be represented by a monotonic function. Much like the Pearson coefficient, a Spearman coefficient score between 0.9 and 1 is acceptable. The Spearman coefficient is given by:

$$S_c = 1 - \frac{6}{N(N^2-1)} \times \sum_{i=1}^{N} d_i(x_i, y_i)^2 \quad (2)$$

where,

$$d_i(x_i, y_i) = rank(x_i) - rank(y_i) \quad (3)$$

being rank (x_i) and rank(y_i) the position the variables x_i and y_i assume in the sorted list of x and y, respectively.

The outlier ratio measures the consistency of the predicted MOS with the subjective MOS. It is given by:

$$Outlier\ ratio = \frac{N_0}{N} \quad (4)$$

where N is the total number of data points and N_0 is the number of outlier data points. An outlier point is a point that does not belong to the interval $[MOSt_k - 2 \times \sigma_k\ , MOSt_k + 2 \times \sigma_k]$, being $MOSt_k$ the subjective MOS of the sequence k and σ_k the standard deviation of sequence k calculated using the subjective test results.

The RMS error measures how much the predicted MOS differ from the subjective MOS. A high difference between the two variables results in a high RMS error. Therefore, if a model scores a high RMS error then there's a strong indicator that the model may be inadequate. The RMS error is given by:

$$\mathrm{RMS} = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{N}} \quad . \quad (5)$$

Chapter 3

Objective Quality Assessment of Encoded Video

3.1 Introduction

As mentioned in the previous chapter, due to DCT coefficients quantization in standard coders, video's perceived quality is reduced when a video is encoded. In this chapter, new objective quality metrics for encoded video are proposed and assessed using the MOS values resulting from subjective tests together with the correlation metrics proposed by VQEG. These algorithms have in common a module, proposed in [BrQu10], which estimates the error due to lossy encoding of the

video signals, using only information derived from the compressed bitstream. In order to obtain perceived quality scores, the estimated error is combined with a video activity index also computed from the bitstream.

This chapter is organized as follows. Since MOS values are used to train and validate the objective quality assessment algorithms proposed in this thesis, we start by describing, in section 3.2, the subjective assessment tests that were conducted to obtain such values. Section 3.3 overviews the rationales behind the development of the objective metrics analyzed along this chapter. Section 3.4 describes the proposed objective quality metrics. Section 3.5 evaluates and compares those metrics with state of the art algorithms developed with the same purpose.

3.2 Subjective quality assessment of encoded video

The main goal of all video quality prediction algorithms is to be able to predict the opinion a human observer would give, when evaluating the video's quality. Therefore, subjective video quality evaluation is essential to benchmark the objective video quality metrics. These subjective evaluations use human participants and specific evaluation methods. The human participants view and evaluate previously selected video sequences. After a statistical analysis of the subjective scores, the Mean Opinion Score (MOS) of the human participants is obtained for every video sequence. The subjective data can then be used to calibrate or to validate the quality prediction algorithms.

The subjective data used in this chapter was obtained through subjective video quality assessment tests performed at *Instituto Superior Técnico* [PQR09] with the purpose of studying the subjective quality of H.264/AVC and MPEG-2 encoded video. The evaluation method used was the Double Stimulus Impairment Scale (DSIS), described in Recommendation ITU-R BT.500-9 [ITU98]. This method consists in presenting, to the observer, video sequences organized in pairs: firstly the original, undistorted video sequence is shown and secondly the encoded, distorted video sequence is also shown. Following the presentation of both video sequences, participants are asked to evaluate the encoded video while having the original video as reference. The evaluation is done by rating the encoded video on a scale of 1 to 5, being: 1 - Very annoying; 2 - Annoying; 3 – Slightly Annoying; 4 – Perceptible, but not annoying; 5 – Imperceptible.

The selected video sequences for these tests are shown in Figure 12. All of the sequences are in CIF format (352x288 pixels), are 10 seconds long and have a frame rate of 30 Hz. When choosing the video sequences, it was taken into account the fact that the video's subjective quality also depends on the video content. Two parameters were used for this purpose: the video spatial activity and the video temporal activity (as defined in [PQR09]). In Figure 13, both the spatial and temporal activities are presented for the selected videos. These video sequences were chosen since they span a broad range of spatial-temporal activities.

Figure 12 - Selected video sequences for the subjective quality tests

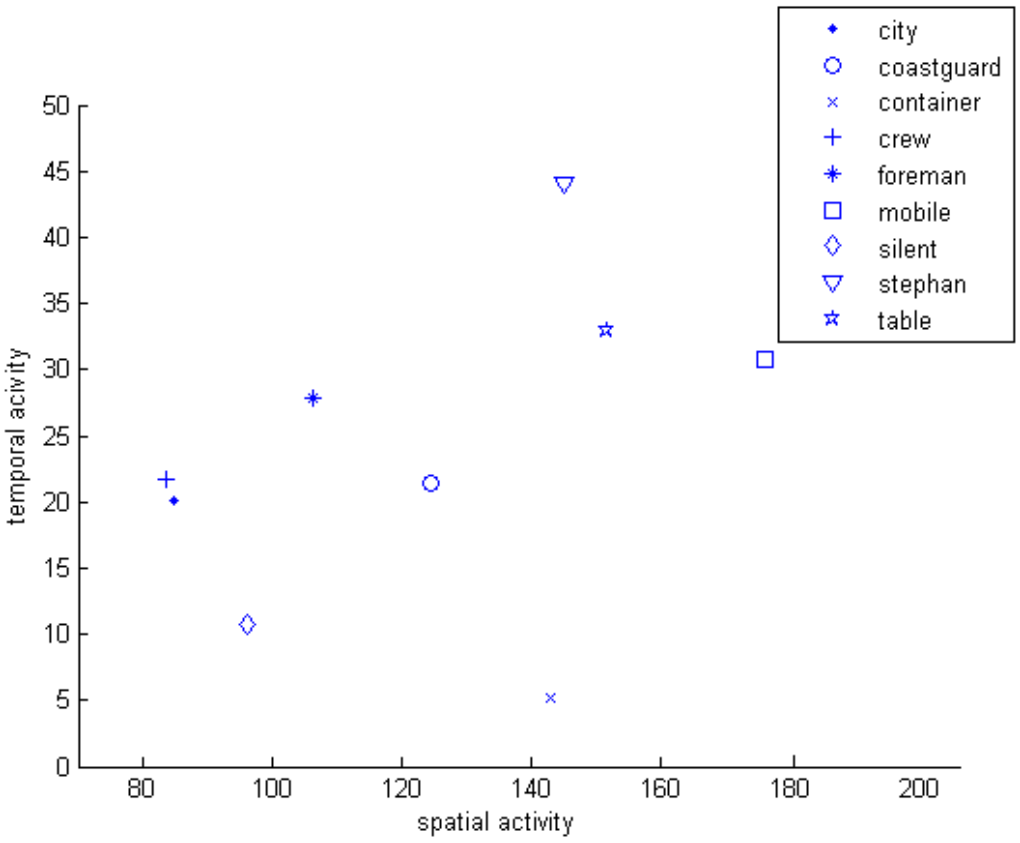


Figure 13 - Spatial-Temporal activity of the selected video sequences

Each selected video sequence is shown to the observers with various video qualities. In order to achieve this, the sequences were compressed with different bit rates. In general, a higher bit rate translates into a higher video quality. Table 1 summarizes the bit rates used to compress the video sequences with the H.264/AVC codec.

Table 1 - Encoding bit rates using H.264/AVC for the selected video sequences

Video Sequence	**Bit Rates [kbit/s]**
City	128; 200; 256; 512
Coastguard	64; 100; 128; 200; 256; 512
Container	64; 128; 256; 512
Crew	128; 200; 400; 1024
Football	256; 400; 512; 750; 1024; 2048
Foreman	64; 128; 256; 512
Mobile	64; 128; 200; 256; 400; 512
Silent	64; 200; 400; 1024
Stephan	128; 200; 256; 400; 512; 1024
Table	64; 128; 256; 512
Tempete	128; 200; 400; 750

After the subjective video quality evaluation, a statistical analysis is conducted in order to remove any outliers (any data outside the confidence interval). After the analysis, the final MOS is obtained for each video sequence concluding the subjective quality assessment.

3.3 Objective quality models for encoded video

As previously mentioned, the subjective quality of a encoded video sequence, expressed through the MOS, depends on how much the video was compressed – video encoded with a high bit rate usually has a better quality (high MOS value) than a video encoded at a lower bit rate. The same behaviour is observed for the mean square error (MSE) of the encoded video. In fact, although the MSE is a rough measure of the perceived quality, its correlation with the MOS tends to be high for the same sequence, when encoded at different bit rates, and using the same encoder. This conclusion, already shown by other authors (e.g., [BRK09], [BRQ09]), may also be confirmed by Figure 14, which shows the MOS versus the MSE values for the different video sequences, and encoding conditions, used in the subjective tests (described in section 3.2). For the same sequence, and with very few exceptions, the plot MOS(MSE) lays in a straight line, which confirms the strong correlation between MOS and MSE measurements.

However, the previous conclusion does not hold when considering different video sequences – in this case, an increase of MSE may not correspond to a decrease in MOS values. For instance, looking once more at Figure 14, we may figure out that an increase of MSE from 80 to 160 may be accompanied either by an increase in MOS from 0.46 (“Foreman”) to 0.63 (“Tempete”), either by a decrease in MOS to 0.18 (“Football”).

As seen in section 3.2 all the video sequences considered in Figure 14 are characterised by different spatio-temporal content; this has a strong influence on the resulting perceived quality, when the videos are compressed with the same compression factors (same bit rate at the encoder output).

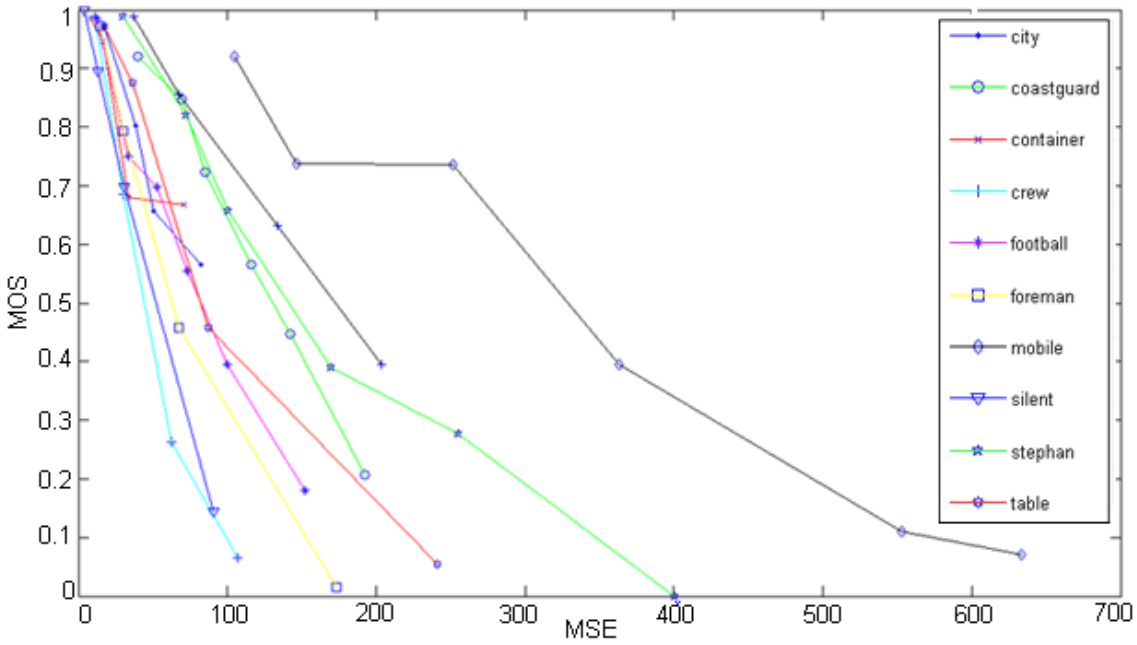


Figure 14 - MOS versus MSE

The MSE is defined as the mean squared difference between the original sequence and the coded sequence. When applied to the video luminance component, it is expressed as:

$$\mathrm{MSE} = \frac{1}{\mathrm{M}\times\mathrm{N}\times\mathrm{T}} \times \sum_{\mathrm{t}=1}^{\mathrm{T}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \sum_{\mathrm{j}=1}^{\mathrm{M}} [\mathrm{Y_o}(\mathrm{i},\mathrm{j},\mathrm{t}) - \mathrm{Y_c}(\mathrm{i},\mathrm{j},\mathrm{t})]^2 \quad (6)$$

where Y_o represents the luminance of the original t-th frame at pixel (i,j), Y_c the luminance of the compressed t-th frame at pixel (i, j), T the total number of frames and M,N the number of pixels per line and the number of lines of each frame, respectively.

With the MSE we can obtain the Peak Signal-to-Noise Ratio (PSNR), which is commonly used as an objective measure of video quality. The PSNR is the ratio between the maximum possible value of luminance (for pixels represented in 8 bit per sample this value is 255) and the MSE, and it is usually expressed in logarithmic units as:

$$\mathrm{PSNR} = 10\ log_{10}\left(\frac{255^2}{MSE}\right), \text{[dB]} \quad (7)$$

The NR objective metrics described and evaluated in this chapter exploit:

- The monotonic behaviour of the MOS versus MSE values, when the same video sequence is considered.
- The dependency of the MOS versus MSE model parameters on the spatial-temporal activities, when different video sequences are considered.

All the developed metrics use the error estimation module proposed in [BrQu10], for a no-reference MSE estimate of the encoded video sequences.

3.4 The MOS prediction models

3.4.1 MOS versus MSE approximation curves

Based on the relationship between MOS and MSE previously mentioned, four different MOS prediction models are studied in this chapter.

The first MOS prediction model was proposed by Bhat in [BRK09] as a FR model and it considers that the relationship between MOS and MSE can be seen as a straight line with slope $-k_s$ and a y-intercept of 1; mathematically, it can be expressed as:

$$\mathrm{MOSp} = 1 - k_s(\mathrm{MSE}) \quad (8)$$

where MOSp is the predicted MOS. By using linear regression of the MOS values, obtained from the subjective test, versus the corresponding MSE values, it is possible to obtain the straight line parameter (k_s value) for each video sequence. Figure 15 shows the subjective data and the straight

line resulting from the linear regression for the “Crew”, “Foreman”, “Mobile” and “Stephan” video sequences.

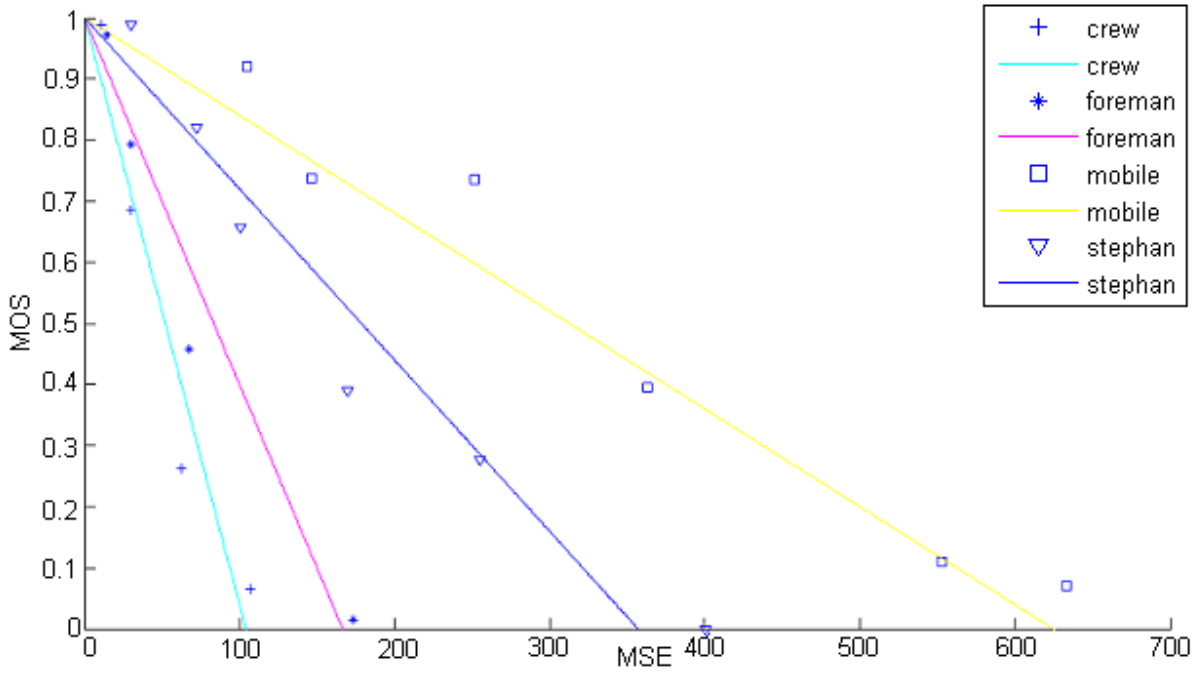


Figure 15 - Regression curves for the linear model for Crew, Foreman, Mobile and Stephan

However, observing Figure 14, it becomes clear that the MOS versus MSE evolution has not the same behaviour for the highest values of MSE. The straight line parameter from the previous model doesn’t seem to be constant and appears to decrease as the MSE increases. In other words, the quality seems to decrease faster on lower MSE values when compared to higher MSE values. Therefore, another possible model is to consider the relation between MSE and MOS as an exponential function, which can be expressed by:

$$\mathrm{MOSp} = \exp\left(-\frac{\mathrm{MSE}}{k_s}\right) \tag{9}$$

where MOSp is the predicted MOS and k_s is the exponential parameter. By using regression with the subjective data and the real MSE values, the exponential parameters were obtained for each sequence. Figure 16 shows the subjective data and the resulting regression curves for the “Crew”, “Foreman”, “Mobile” and “Stephan” sequences.

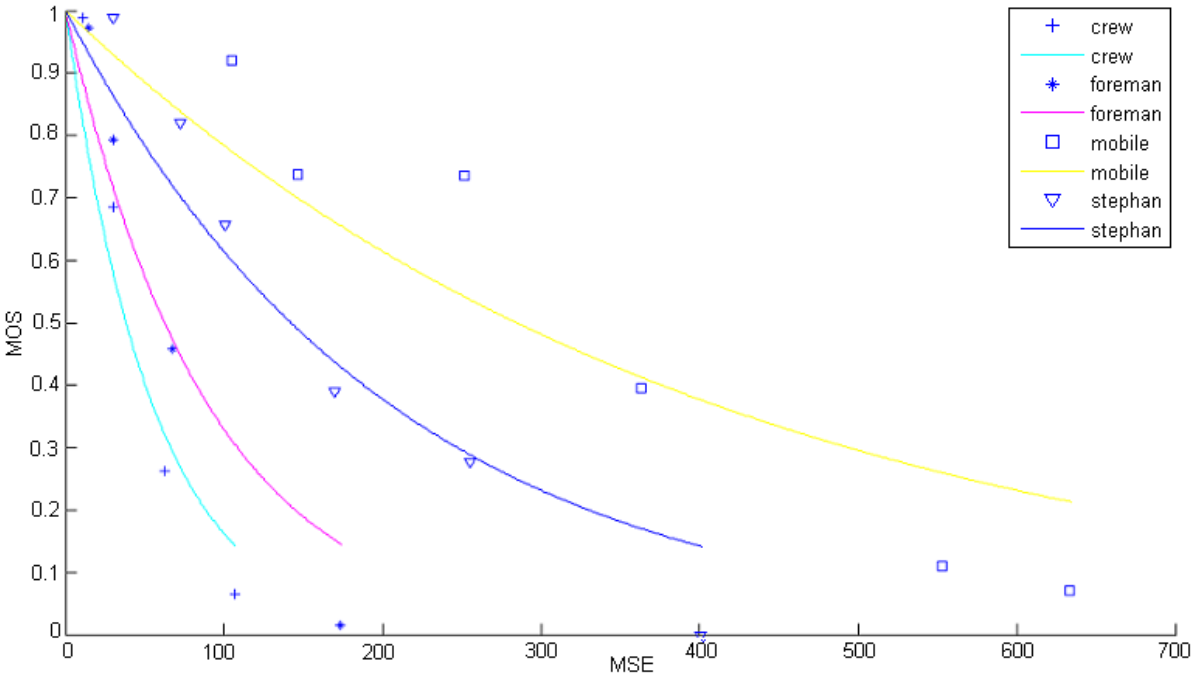


Figure 16 - Regression curves for the exponential model for Crew, Foreman, Mobile and Stephan

By taking a closer look at Figure 14, we can see that the MOS versus MSE relation is not a simple exponential function, as considered in the previous model. In fact, it seems to resemble a sigmoid function since the quality has a slower decrease on lower and higher MSE values when compared to mid MSE values. This third model (Sigmoid1 model) uses a sigmoid function which can be expressed as:

$$\mathrm{MOSp} = \frac{1+e^{-k_1 k_2}}{1+e^{k_1(\mathrm{MSE}-k_2)}} \tag{10}$$

where MOSp is the predicted MOS and k_1 and k_2 are the sigmoid parameters. It takes into account that, for the same video sequence, the quality has a slower decay on lower and higher MSE values, when compared to mid MSE values. By using, once again, regression with the subjective data and the real MSE values, it is possible to obtain the sigmoid parameters for each sequence. Figure 17 shows the subjective data and the regression curves obtained for the “Crew”, “Foreman”, “Mobile” and “Stephan” sequences.

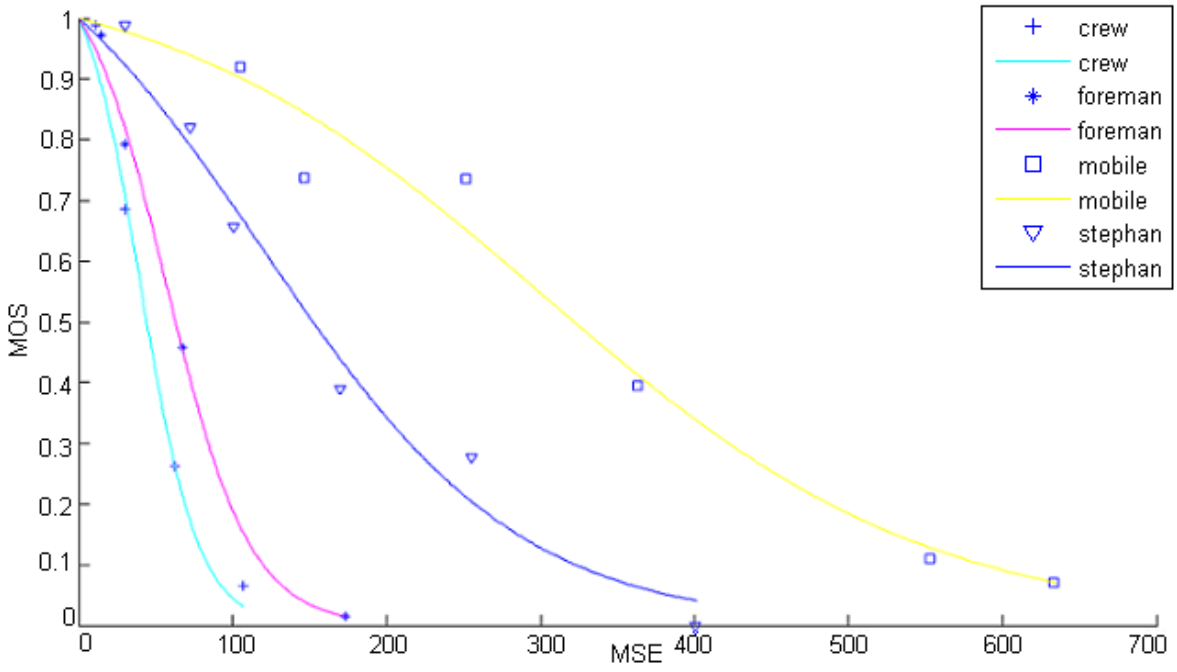


Figure 17 – Regression curves for the sigmoid model (MSE) for Crew, Foreman, Mobile and Stephan

The fourth and final model was proposed in [WP02] as a FR model and it also follows a sigmoid function (Sigmoid2 model). However, it uses the PSNR measurement to estimate the MOS. Mathematically, this model is expressed as:

$$\mathrm{MOSp} = 1 - \frac{1}{1+e^{k_1(\mathrm{PSNR}-k_2)}} \tag{11}$$

where MOSp is the predicted MOS and k_1 and k_2 are the sigmoid parameters. Applying, once again, regression with the subjective data and the real PSNR values, the sigmoid parameters were obtained for each sequence. Figure 18 shows the subjective data versus MSE (for a better comparison with the previous models, the plot is versus MSE and not PSNR values)) and the regression curves obtained for the “Crew”, “Foreman”, “Mobile” and “Stephan” sequences.

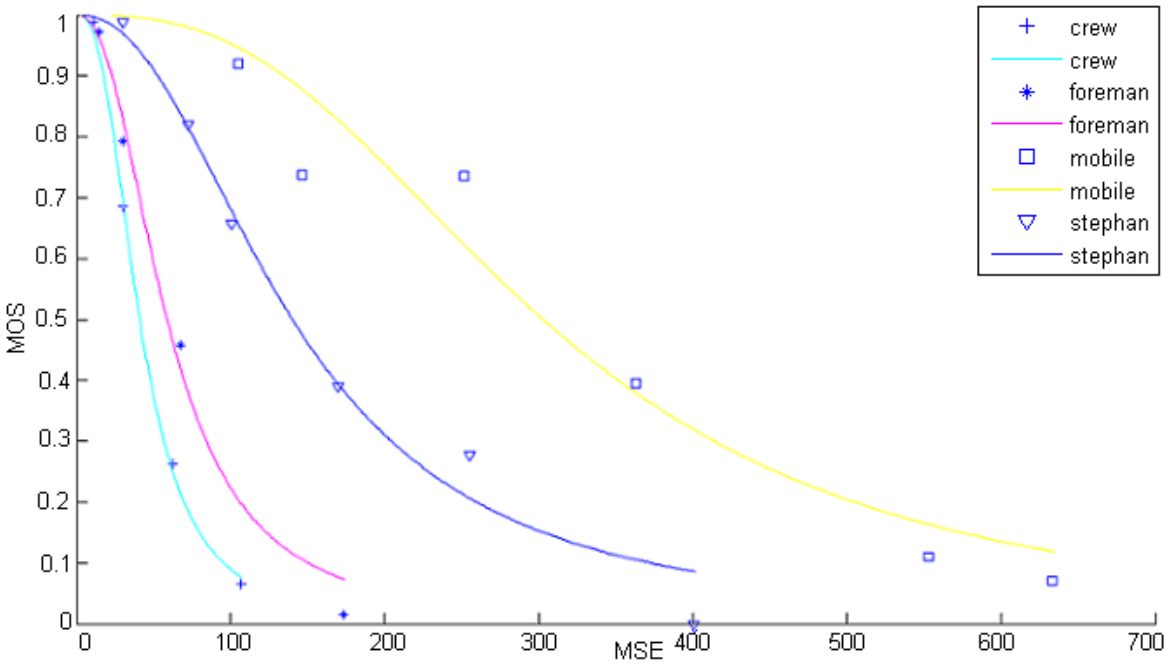


Figure 18 - Regression curves for the sigmoid model (PSNR) for Crew, Foreman, Mobile and Stephan

3.4.2 Estimation of the model parameters

In the previous section we have seen that each model predicts the MOS by using the MSE and one or two parameters; these parameters were obtained by regression using the subjective data (MOS values). However, in a practical transmission scenario the subjective data is unavailable, so those parameters have to be estimated from the received data (video bitstream and/or decoded video). In [BRK09], where the straight line model was proposed, the authors showed that the required parameter, k_s, is related to the video content activity. This approach was also followed in this thesis, in order to estimate the model parameters required by the new models. The definition of video activity and how it correlates with the model parameters will be analyzed in the following sub-sections.

3.4.2.1 *Definition of video sequence activity*

The video activity is typically characterized through its spatial and temporal activities, and the scientific literature provides several different methods of measuring these activities. In this thesis, the methods recommended in [BRK09] have been used.

A video sequence with a high spatial activity is a sequence rich in texture, where compression artefacts are better masked. The spatial activity measurement is computed using the two Sobel filters shown in Figure 19. One is used to determine the horizontal gradient (G_h) while the other is used to determine the vertical gradient (G_v). In order to obtain for each pixel a single measure, the gradient norm (the square root of the sum of the vertical and horizontal gradient squares), $G(x,y)$, is computed:

$$G(x,y) = \sqrt{G_h(x,y)^2 \times G_v(x,y)^2} \tag{12}$$

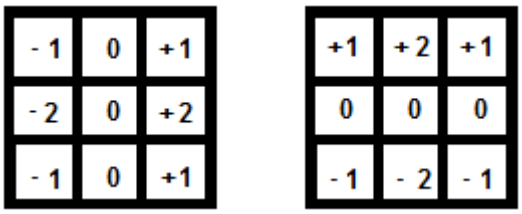


Figure 19 - Sobel Filters

After obtaining $G(x,y)$ for every pixel, its average value if computed. This is done for each frame, resulting in a vector containing the spatial activity along the video sequence. Finally, the mean value of the vector is computed resulting in the global spatial activity of the sequence.

A video sequence with high temporal activity is a sequence with large temporal changes. In these conditions, an artefact can become unnoticed by the viewer. The temporal activity is obtained by first computing the absolute difference (Y_{diff}) between each two successive frames:

$$Y_{diff} = \left| Y_{currentframe} - Y_{previousframe} \right| \tag{13}$$

Then, using the two Sobel filters, the horizontal and vertical gradients (G_h and G_v) are determined for the resulting Y_{diff}. The gradient norm for each pixel is then obtained through (12). After obtaining $G(x,y)$ for every pixel, its average value is computed. This is done for each frame difference, resulting in a vector containing the temporal activity along the sequence. Finally, the mean value of the vector is computed, resulting in the global temporal activity of the sequence.

According to [BRK09] the global activity of the video sequence is then defined as the maximum between the spatial and temporal activities. Table 2 shows the activities computed for all the eleven video sequences. It can be seen that, for the considered set of video sequences, the spatial activity is always higher than the temporal activity. This result may be due to the fact that both activities have different ranges and normalization may be required before comparing them. However, in this thesis, we have considered the global activity to be simply the global spatial activity.

Table 2 - Activities for the original test sequences

Video	*Spatial activity*	*Temporal activity*	*Video Activity*
City	*73.6*	*40.5*	*73.6*
Coastguard	*98.2*	*38.6*	*98.2*
Container	*82.7*	*6.4*	*82.7*
Crew	*46.1*	*31.4*	*46.1*
Football	*69.9*	*54.1*	*69.9*
Foreman	*61.8*	*30.3*	*61.8*
Mobile	*150.8*	*59.7*	*150.8*
Silent	*60.9*	*9.5*	*60.9*
Stephan	*122.5*	*80.3*	*122.5*
Table	*82.0*	*22.7*	*82.0*
Tempete	*101.8*	*30.2*	*101.8*

The defined video activity requires the decoded video sequence. However, an estimation of the video spatial activity could also be obtained with information taken from the bitstream, namely the DCT coefficients. We propose to compute the video activity by first estimating the spatial activity of each I-frame through:

$$\sigma_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(DCTcoef_i - \mu)} \quad (14)$$

where σ_j is the estimated spatial activity of the j-th I-frame, $DCTcoef_i$ is the i-th DCT coefficient of the frame, μ is the average value of the DCT coefficients in the frame and n is the number of DCT coefficients in the frame.

The estimated activity of the video results from the average of the spatial activity of all I-frames:

$$Estimated\ activity = \frac{1}{N}\sum_{j=1}^{N}\sigma_j \quad (15)$$

where N is the number of I frames and σ_j is the estimated spatial activity of the j-th I frame.

In order to validate the estimated activities, the Pearson correlation between them and the activity values computed in the pixel domain, using the original video (shown in Table 2) was computed, and a value of 0.97 was obtained. It was also verified that the effect of compression as a marginal impact on this value.

3.4.2.2 *Estimation of the model parameters using the video activity*

As previously mentioned, the quality model parameters vary with the contents of the video sequence. In this sub-section, we analyse the relation between the video activity and the parameters of each model.

In the following, the MSE values can be obtained directly from the original and encoded videos (true MSE) or using the no-reference MSE estimate from [BrQu10] (estimated MSE). As for the video activity, it can be obtained directly from the uncompressed videos (true activity) or by using equations (14) and (15) (estimated activity).

Figure 20 shows how the model parameters, obtained using the true MSE values, relate with the video activity; this figure suggests that some model parameters have a linear relation with the video activity, while others have an exponential relation.

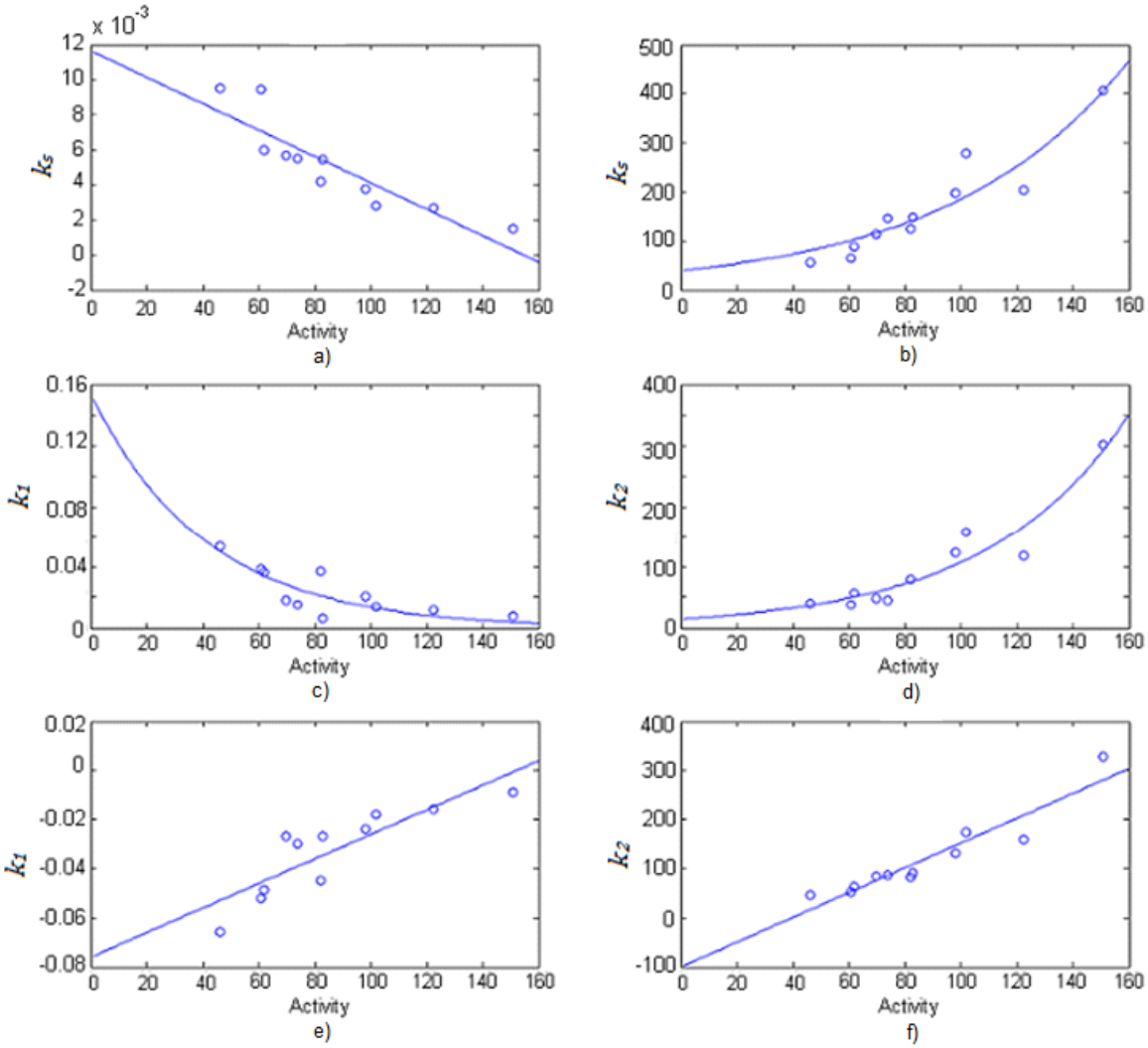


Figure 20 - Model parameters versus video activity, and resulting regression curves, for: a) linear model, b) exponential model, c) and d) sigmoid1 model, e) and f) sigmoid2 model, using the true MSE and video activity values.

The parameters (k_s) for the exponential and the sigmoid1 models were considered to have an exponential relation with the video activity, which can be expressed as:

$$model\ parameter = \beta_1 \times \exp(Activity \times \beta_2) \quad (16)$$

The parameters (k_s) for the linear and sigmoid2 models were considered to have a linear relation with the video activity, which can be expressed as:

$$model\ parameter = \beta_1 \times Activity + \beta_2 \quad (17)$$

Parameters β in (16) and (17) are obtained by regression using the subjective MOS, the MSE values, and the video activity, by substituting (16) or (17) in (8), (9), (10) and (11). The resulting fitting curves are represented in Figure 20 (using true MSE and video activity values) and Figure 21 (using

estimated MSE and video activity values computed from the compressed video). The functions used in these regressions are shown in Table 3, while Table 4 shows the resulting β values (using true MSE and video activity) when all the video sequences are used to train the models.

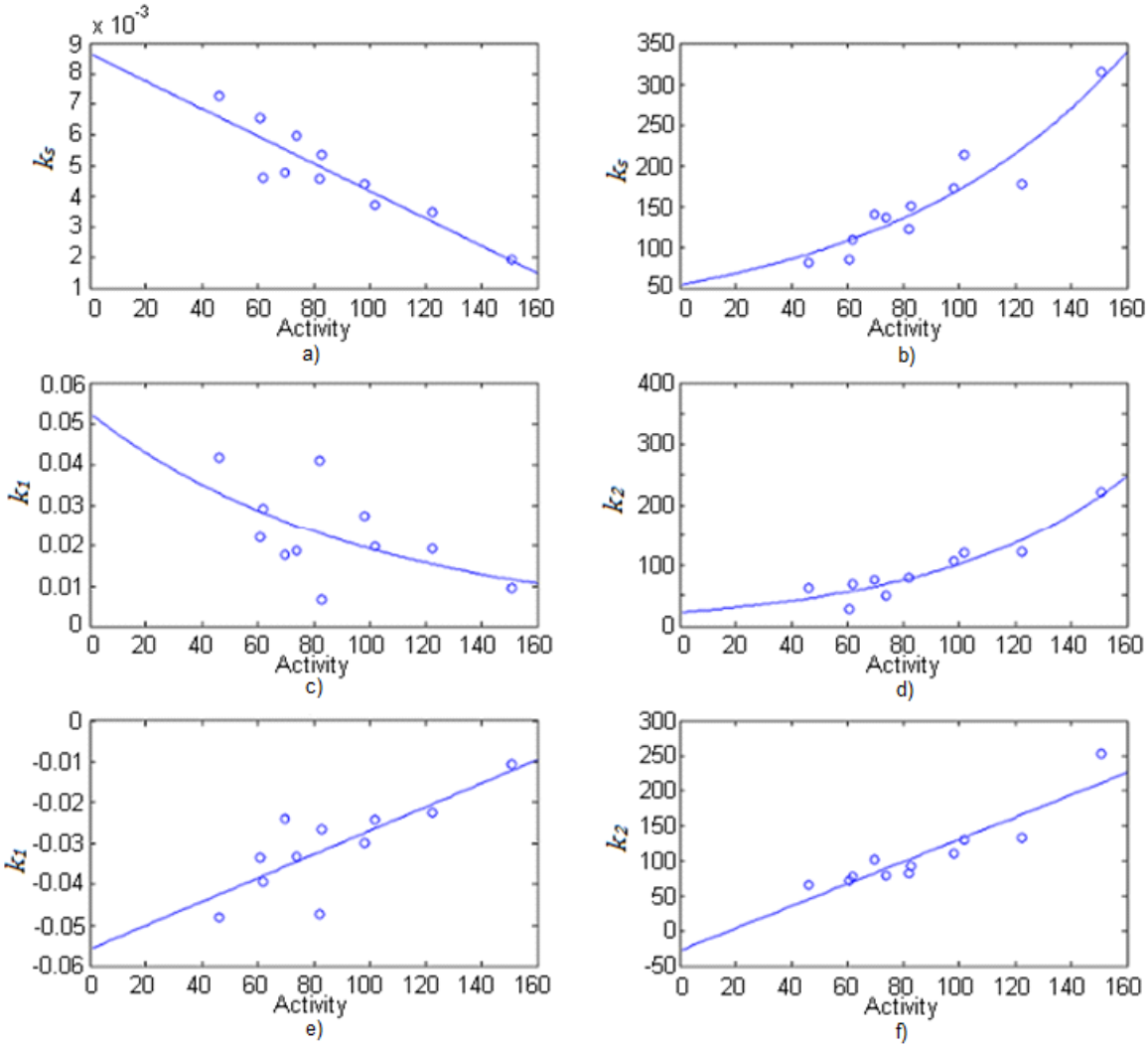


Figure 21 - Model parameters versus estimated video activity, and resulting regression curves, for: a) linear model, b) exponential mode, c) and d) sigmoid1 model, e) and f) sigmoid2 model, using the estimated MSE and video activity values

Table 3 - Complete function for the four prediction models

Model	***Function***
Linear	$MOS = 1 - (\beta_1 \times Activity + \beta_2) \times MSE$
Exponential	$MOS = exp\left(-\frac{MSE}{\beta_1 . exp\ (Activity \times \beta_2)}\right)$
Sigmoid1	$MOS = \frac{1 + exp\ (-(\beta_1 . exp\ (Activity \times \beta_2) \times (\beta_3 . exp\ (Activity \times \beta_4))}{1 + exp\ [(\beta_1 . exp\ (Activity \times \beta_2) . (MSE - (\beta_3 . exp\ (Activity \times \beta_4))]}$
Sigmoid2	$MOS = 1 - \frac{1}{1 + exp\ [(\beta_1 \times Activity + \beta_2) . (PSNR - (\beta_3 \times Activity + \beta_4))]}$

Table 4 - Values of the parameters β using all sequences as training sequences

Model	***β***			
Linear	*β_1= -0.0001*		*β_2= 0.009*	
Exponential	*β_1= 31.620*		*β_2= 0.017*	
Sigmoid1	*β_1= 0.086*	*β_2= 0.017*	*β_3= 16.750*	*β_4= 0.019*
Sigmoid2	*β_1= 0.0003*	*β_2= 0.189*	*β_3= -0.180*	*β_4= 81.002*

3.4.3 Predicting the MOS

Now the MOS prediction for each video sequence can be done by using the described NR prediction models. To do so it is required:

- The functions of the prediction models, shown in Table 3.
- A training video set to train the models by calculating the parameters β.
- An estimated MSE value of the video sequence whose MOS we want to predict [BrQu10].
- An estimated (eq. (15)) of the activity for the video sequence whose MOS we want to predict.

To validate objective quality metrics, it is often used a training set and a validation set. The training set calibrates the metrics which, in our case, correspond to the finding of the β values; the

validation set is used to evaluate the metrics. Since we have a small set of video sequences, the leave-one-out cross-validation was used. This method is done by turns, in each turn the different encoded versions of the same video sequence are used as the validation set while all the other video sequences are used as the training set. In each turn, the MOSp of the validation set are obtained. This is repeated until all the video sequences have been used as the validation set and the corresponding MOSp obtained.

After performing the leave-one-out cross-validation method on the eleven video sequences, their MOSp was calculated for each prediction model. Figure 22 and Figure 23 show the MOS versus MOSp for the different models using, respectively, the true and estimated MSE and video activities.

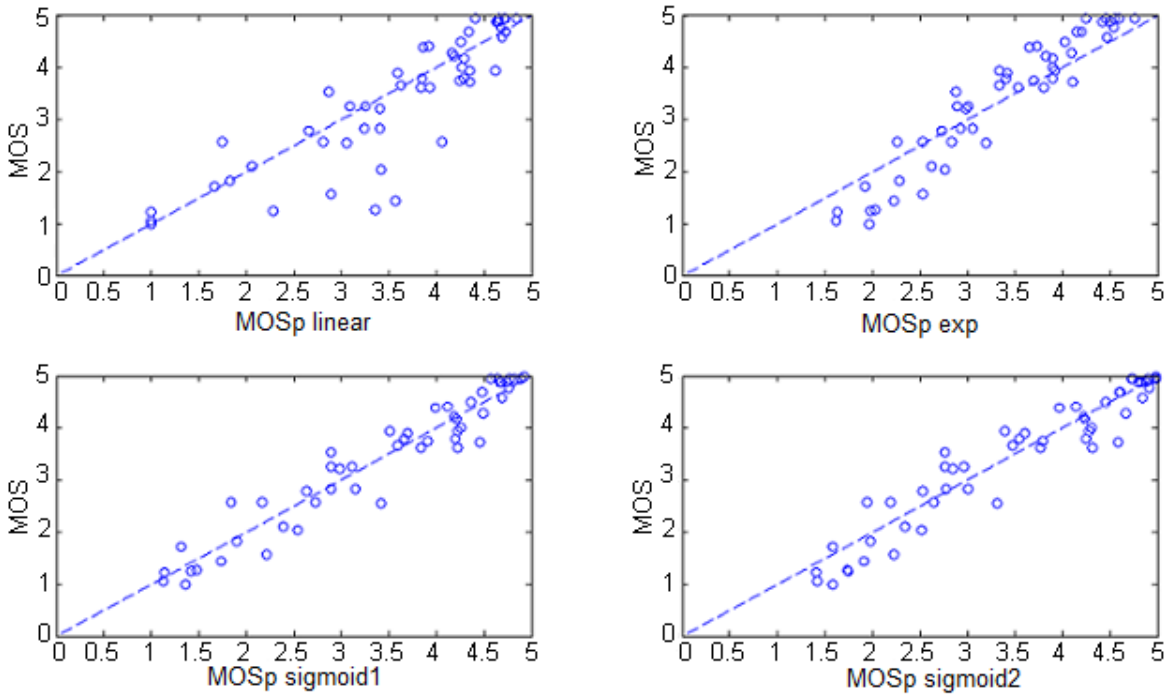


Figure 22 - MOSp versus MOS for the four prediction models using true MSE and true activity

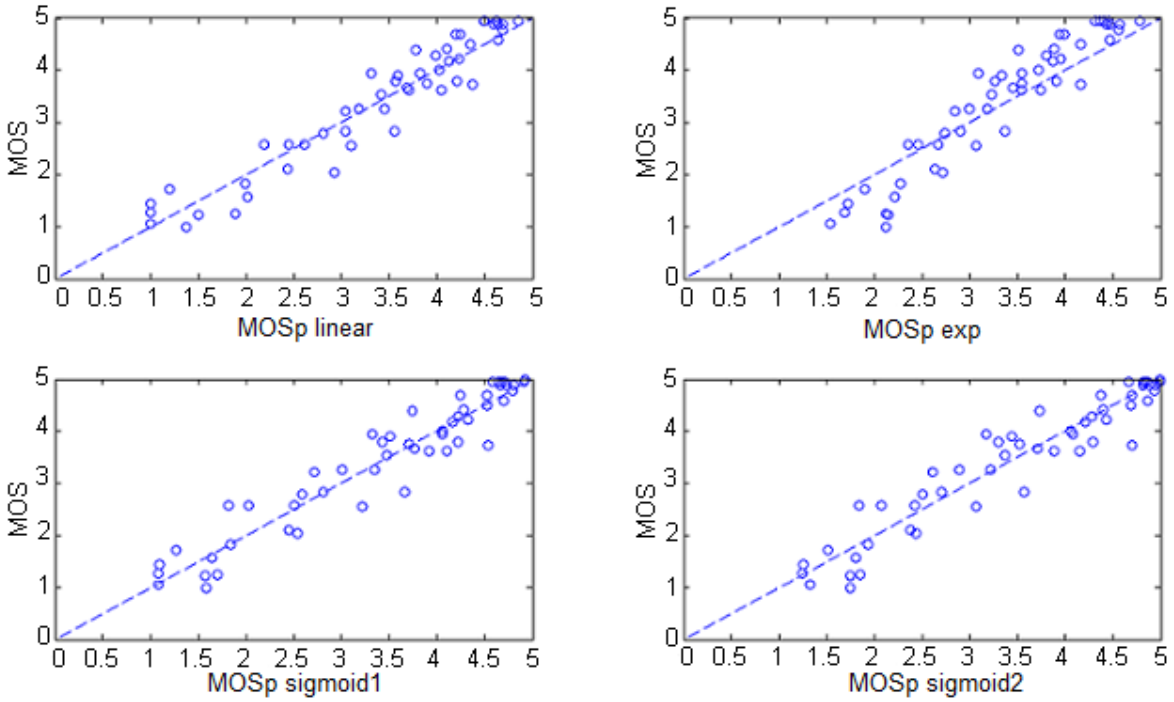


Figure 23 - MOSp versus MOS for the four NR prediction models using estimated MSE and estimated activity

3.5 Results and model comparison

To compare the four models, the VQEG performance metrics described in section 2.6, namely Pearson and Spearman correlation coefficients and root mean squared (RMS) error, were used. First, the comparison is done using the models with the true MSE and video activities. Table 5 and Table 6 show the Pearson coefficient and the RMS, respectively, for each individual video, while Table 7 shows the three performance metrics obtained with all videos.

Table 5 - Pearson coefficients using true MSE and true activity for each individual video

Video	***Pearson***			
	Linear	*Exponential*	*Sigmoid1*	*Sigmoid2*
City	*0.959*	*0.979*	*0.945*	*0.959*
Coastguard	*0.996*	*0.984*	*0.996*	*0.984*
Container	*0.851*	*0.878*	*0.814*	*0.777*
Crew	*0.963*	*0.994*	*0.989*	*0.999*
Football	*0.986*	*0.992*	*0.989*	*0.983*
Foreman	*0.978*	*1.000*	*0.998*	*0.997*
Mobile	*0.981*	*0.976*	*0.983*	*0.970*
Silent	*0.999*	*0.998*	*0.996*	*0.995*
Stephan	*0.983*	*0.995*	*0.975*	*0.983*
Table	*0.981*	*0.995*	*0.992*	*0.999*
Tempete	*0.999*	*0.995*	*0.998*	*0.993*

Table 6 - RMS using true MSE and true activity for each individual video

Video	***RMS***			
	Linear	*Exponential*	*Sigmoid1*	*Sigmoid2*
City	*0.197*	*0.356*	*0.234*	*0.259*
Coastguard	*0.298*	*0.468*	*0.219*	*0.273*
Container	*0.324*	*0.297*	*0.372*	*0.449*
Crew	*0.897*	*0.563*	*0.309*	*0.355*
Football	*0.188*	*0.262*	*0.289*	*0.277*
Foreman	*0.319*	*0.391*	*0.122*	*0.189*
Mobile	*1.397*	*0.554*	*0.275*	*0.374*
Silent	*0.709*	*0.487*	*0.384*	*0.416*
Stephan	*0.254*	*0.549*	*0.474*	*0.512*
Table	*0.290*	*0.388*	*0.192*	*0.130*
Tempete	*0.648*	*0.610*	*0.545*	*0.529*

Table 7 - Correlation coefficients using true MSE and true activity for all videos

Correlation Coefficient	***Model***	***All***
RMS	*Linear*	*0.636*
	Exponential	*0.463*
	Sigmoid1	*0.332*
	Sigmoid2	*0.365*
Pearson	*Linear*	*0.868*
	Exponential	*0.958*
	Sigmoid1	*0.963*
	Sigmoid2	*0.956*
Spearman	*Linear*	*0.904*
	Exponential	*0.957*
	Sigmoid1	*0.956*
	Sigmoid2	*0.953*
Outlier	*Linear*	*0.115*
	Exponential	*0.096*
	Sigmoid1	*0.019*
	Sigmoid2	*0.038*

The same procedure was repeated but now using the estimated MSE and video activities. Table 8 and Table 9 show the Pearson coefficient and the RMS, respectively, for each individual video, while Table 10 shows the three performance metrics obtained with all eleven videos.

Table 8 - Pearson coefficients using estimated MSE and estimated activity for each individual video

Video	***Pearson***			
	Linear	***Exponential***	***Sigmoid1***	***Sigmoid2***
City	*0.955*	*0.972*	*0.937*	*0.934*
Coastguard	*0.994*	*0.988*	*0.995*	*0.985*
Container	*0.852*	*0.877*	*0.811*	*0.756*
Crew	*0.985*	*0.997*	*0.997*	*0.997*
Football	*0.990*	*0.990*	*0.990*	*0.979*
Foreman	*0.993*	*0.999*	*0.999*	*0.997*
Mobile	*0.993*	*0.984*	*0.979*	*0.964*
Silent	*0.993*	*0.999*	*0.990*	*0.990*
Stephan	*0.986*	*0.995*	*0.988*	*0.995*
Table	*0.974*	*0.994*	*0.976*	*0.990*
Tempete	*0.999*	*0.998*	*0.998*	*0.999*

Table 9 - RMS using estimated MSE and estimated activity for each individual video

Video	***RMS***			
	Linear	***Exponential***	***Sigmoid1***	***Sigmoid2***
City	*0.260*	*0.257*	*0.262*	*0.284*
Coastguard	*0.346*	*0.563*	*0.377*	*0.399*
Container	*0.334*	*0.284*	*0.411*	*0.496*
Crew	*0.569*	*0.615*	*0.378*	*0.396*
Football	*0.281*	*0.357*	*0.399*	*0.393*
Foreman	*0.183*	*0.364*	*0.113*	*0.155*
Mobile	*0.411*	*0.527*	*0.438*	*0.445*
Silent	*0.307*	*0.335*	*0.227*	*0.312*
Stephan	*0.374*	*0.627*	*0.419*	*0.401*
Table	*0.410*	*0.582*	*0.450*	*0.465*
Tempete	*0.281*	*0.441*	*0.171*	*0.128*

Table 10 - Correlation coefficients using estimated MSE and estimated activity for all videos

Correlation Coefficient	*Model*	*All*
RMS	*Linear*	*0.355*
	Exponential	*0.478*
	Sigmoid1	*0.359*
	Sigmoid2	*0.377*
Pearson	*Linear*	*0.959*
	Exponential	*0.954*
	Sigmoid1	*0.957*
	Sigmoid2	*0.953*
Spearman	*Linear*	*0.947*
	Exponential	*0.945*
	Sigmoid1	*0.947*
	Sigmoid2	*0.943*
Outlier	*Linear*	*0.038*
	Exponential	*0.154*
	Sigmoid1	*0.038*
	Sigmoid2	*0.038*

In Table 10 we have highlighted in green the model with the best results for each performance metric. The results show that, when the true MSE and true activity is used, the Sigmoid1 model is the one with best results, except in the Spearman coefficient, where the exponential model is slightly better. When the estimated MSE and estimated activity are used, the linear model is the one with best results in the RMS and Pearson coefficient while the Sigmoid1 model has the best Spearman

coefficient. All the models produce very similar results, making them all acceptable. However, for a NR approach the linear model is the best of the four, not due to the correlation coefficients obtained but also because of its lower complexity when compared to the other models.

3.6 Conclusion

In this chapter, four quality prediction models, for compressed video, have been described, analyzed and compared. All models predict the video quality based on its MSE and spatial activity. Eleven video sequences were considered, each one encoded with the H.264/AVC standard and using different bitrates, making a total of 52 video sequences available for training and testing the different prediction models. The models have been analyzed using the true and the estimated MSE and video activities. The results obtained show that the prediction model with the best performance is the linear model. This model was proposed by Bhat in [BRK09] as a FR model (i.e., assuming that the reference video is available for the MSE computation) and estimates the video activity from the reference video and using a pixel domain approach. However, in this thesis it was modified to work as a bitstream based, NR model. This was possible by estimating the video activity with information taken from the bitstream, namely the DCT coefficients, and the error estimation module proposed in [BrQu10].

Chapter 4

Objective Video Quality Assessment in IP Networks

4.1 Introduction

For video quality assessment in IP networks, not only compression losses should be considered, but also transmission losses that might occur. Losses in IP networks may happen for various reasons (as described in [KGPL06]) and may affect video's perceived quality due to different factors. As an example, a loss in an I-frame is expected to have a higher impact on the perceived quality than a loss in a B-frame since, during the encoding process, I-frames are used as reference for

a higher number of frames, than B-frames.

This chapter is organized as follows. Section 4.2 describes the subjective quality assessment tests using H.264/AVC encoded video, corrupted with packet losses. Section 4.3 overviews the main ideas behind the objective metrics developed in the chapter. Section 4.4 describes and analyses a simple video quality model (VQM) based on the ITU-T Rec. G.1070. Section 4.5 proposes and evaluates some modifications of the simple VQM, by taking into consideration various factors, such as frame type, error propagation and video temporal activity, which may impact the video perceived quality. Section 4.6 describes and analyses a statistical VQM which is based on statistical measurements taken from the packet loss pattern. Section 4.7 evaluates and compares the developed VQMs. Finally, section 4.8 synthesises the main conclusions obtained along this chapter.

4.2 Subjective video quality assessment in IP Networks

Much like the previous chapter, subjective data is essential for the development of objective video quality metrics, since it is used for training and testing. The subjective data used in this chapter was obtained through subjective tests performed in *Politecnico di Milano* (PoliMi) – Italy and *Ecole Polytechnique Fédérale de Lausanne* (EPFL) - Switzerland [SNTD09]. The subjective tests addressed the effect of packet loss es on a video's perceived quality when encoded with H.264/AVC. For this purpose, six video sequences in CIF format and with a frame rate of 30 fps, were considered, namely "Foreman", "Hall", "Mobile", "Mother", "News" and "Paris" (shown in Figure 24).

Figure 24 – Sequences a) "Foreman" b) "Hall" c) "Mobile" d) "Mother" e) "News" f) "Paris"

These sequences were selected since they represent various levels of spatial and temporal complexity. The analysis of the content was performed by evaluating the Spatial Information (SI) and Temporal Information (TI) as described in [ITUT99]. The compressed bitstreams were obtained using the H.264/AVC High Profile and the encoding parameters listed in Table 11.

Table 11 - H.264/AVC encoding parameters

Reference software	JM14.2
Profile	High
Number of frames	298
Chroma format	4:2:0
GOP size	16
GOP structure	IBBPBBPBBPBBPBB ...
Number of reference frames	5
Slice mode	Fixed number of MBs
Rate control	Disabled, fixed Quantization Parameter
MB partitioning for motion estimation	Enabled
Motion estimation algorithm	Enhanced Predictive Zonal Search (EPZS)
Early skip detection	Enabled
Selective intra mode decision	Enabled

With these parameters, each frame is split in a fixed number of *slices* (18 *slices*), where each *slice* consists of a full row of MBs, as represented in Figure 25.

Figure 25 - A frame split in 18 *slices*

In the NAL, the bitstreams were formatted for IP networks and with each packet just containing the information of a single *slice.* Therefore, if a packet is lost, a full *slice* is lost.

To simulate the losses, for each H.264/AVC bitstream (each bitstream resulting from the encoding of one of the video sequences presented in Figure 24), a number of corrupted versions were generated by dropping packets. This was done by using error patterns with six different packet loss rates (PLR): 0.1%, 0.4%, 1%, 3%, 5% and 10%. Additionally, two channel realizations were considered for each PLR, resulting in twelve corrupted bitstreams for each of the six selected sequences. Concerning error resilience techniques, it should be noted that neither flexible macroblock address (FMO) nor Intra-frame refreshing, were used. As a result, error propagation is expected to happen. As for error concealment techniques, whenever a packet was lost the decoder used intra-frame prediction on I-frames and inter-frame prediction on P and B-frames

The evaluation method used was the Single Stimulus (SS), which consists in presenting one sequence at a time, without using, as reference, the packet loss free version. A total of 78 sequences were evaluated by each observer, using the 5 point ITU continuous scale [ITUT99] in the range [0-5], shown in Figure 26.

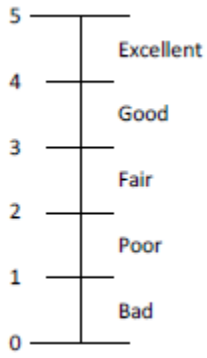


Figure 26 - Five point continuous quality scale

After collecting the scores from the observers, the scores were processed in order to normalize their values and remove any possible outliers. Finally, the MOS values, required to train and validate the objective metrics, were obtained. Since the subjective tests were independently performed by two institutes, we have available two MOS databases - from EPFL and from PoliMi. Figure 27 to Figure 32 (extracted from [SNTD09]) compare the MOS and the confidence intervals obtained from both databases. It can be noticed that the MOS from PoliMi are slightly more optimistic. Additionally, there are a few sequences in which the difference between the obtained MOS from both databases, reaches almost one MOS value.

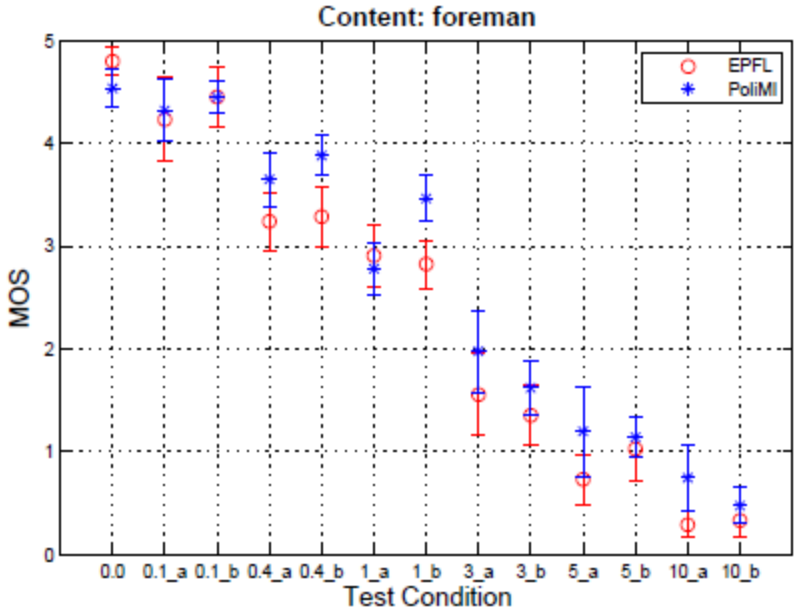


Figure 27 - MOS values from the two databases for Foreman sequence (extracted from [SNTD09])

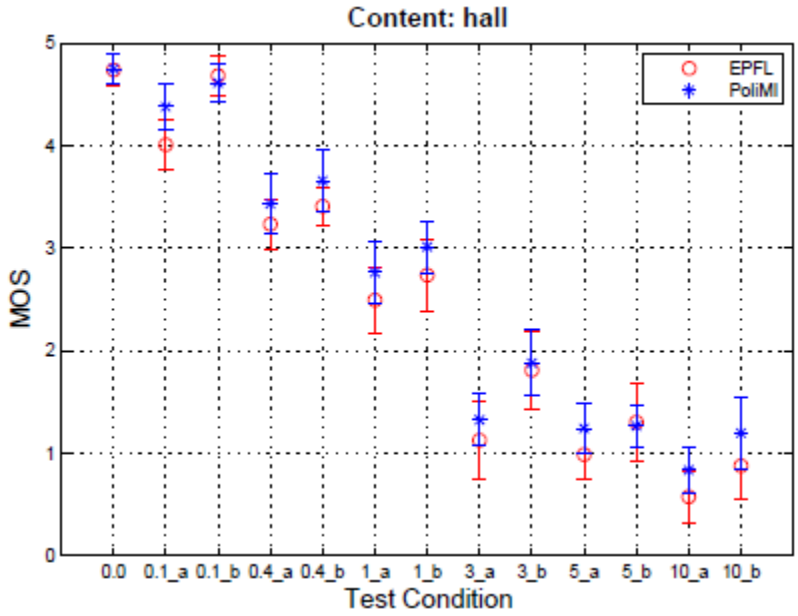


Figure 28 - MOS values from the two databases for Hall sequence (extracted from [SNTD09])

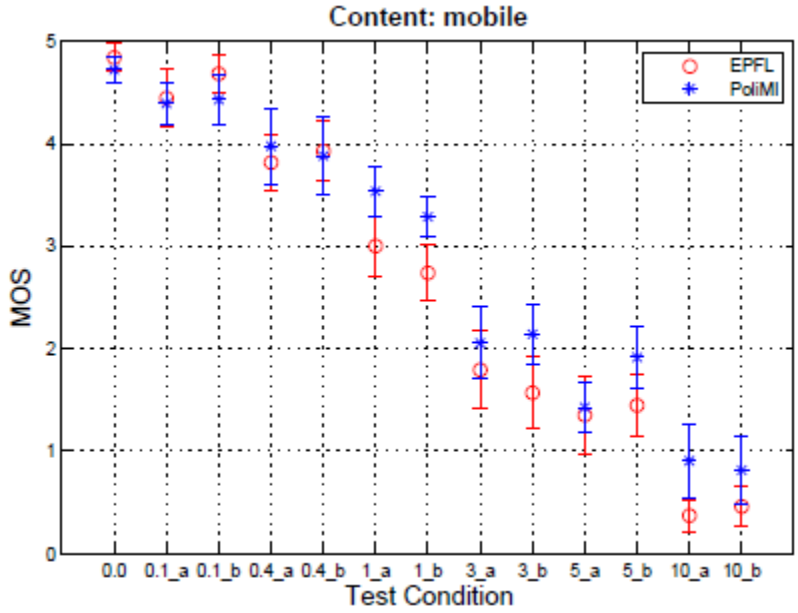


Figure 29 - MOS values from the two databases for Mobile sequence (extracted from [SNTD09])

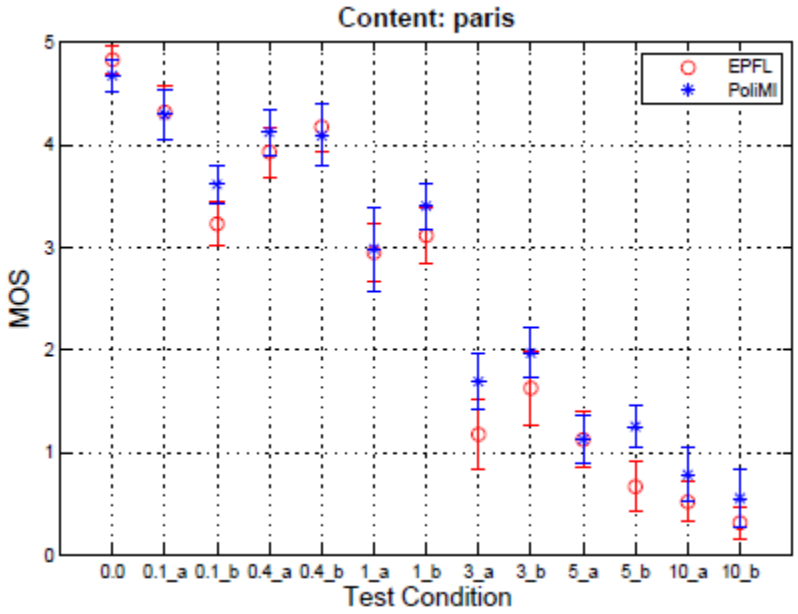


Figure 30 - MOS values from the two databases for Paris sequence (extracted from [SNTD09])

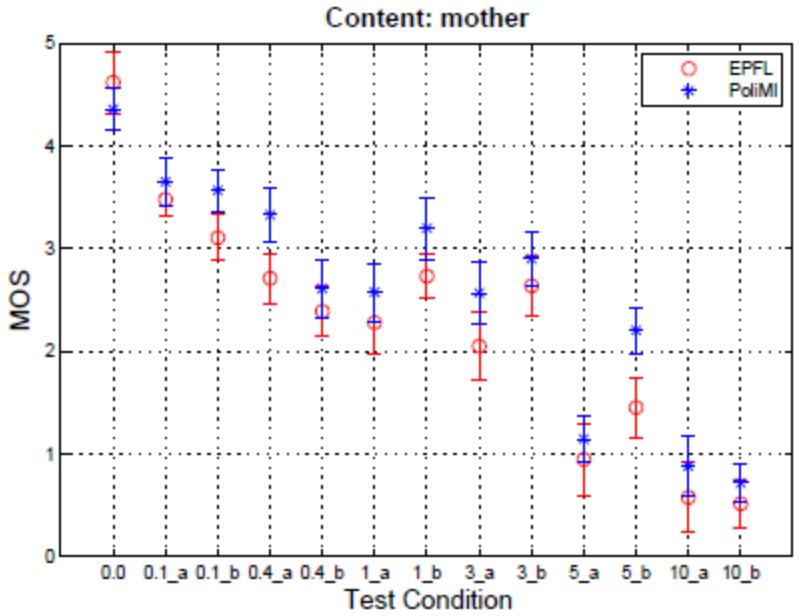


Figure 31 - MOS values from the two databases for Mother sequence (extracted from [SNTD09])

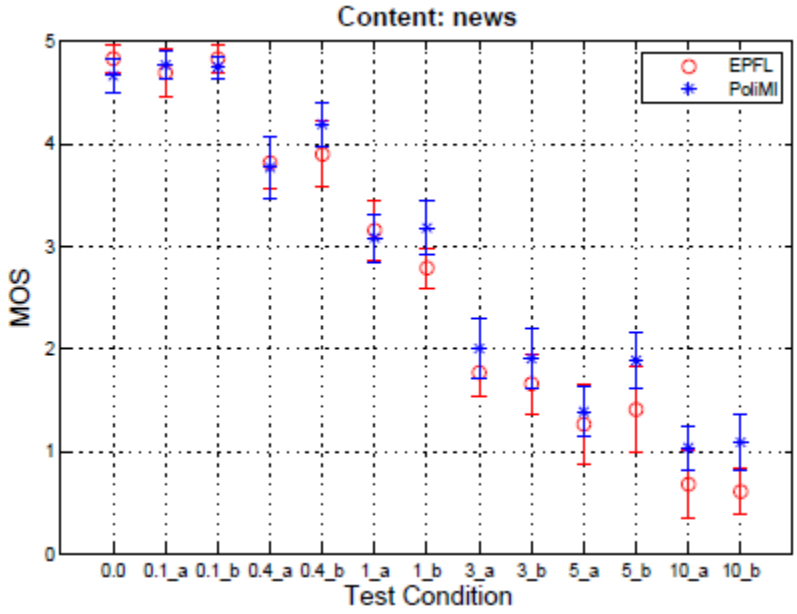


Figure 32 - MOS values from the two databases for News sequence (extracted from [SNTD09])

4.3 Objective quality models for transmission with packet losses

The subjective tests described in the previous section have considered transmission errors (i.e., packet losses). The MOS and the MSE values of each video obtained after decoding, allow the analysis of the MOS versus MSE behavior resulting from packet losses. By looking at Figure 33 to Figure 36, it is possible to conclude that, as in the previous chapter where only losses due to compression were considered, the MOS does not correlate well with the MSE, if we consider all video sequences.

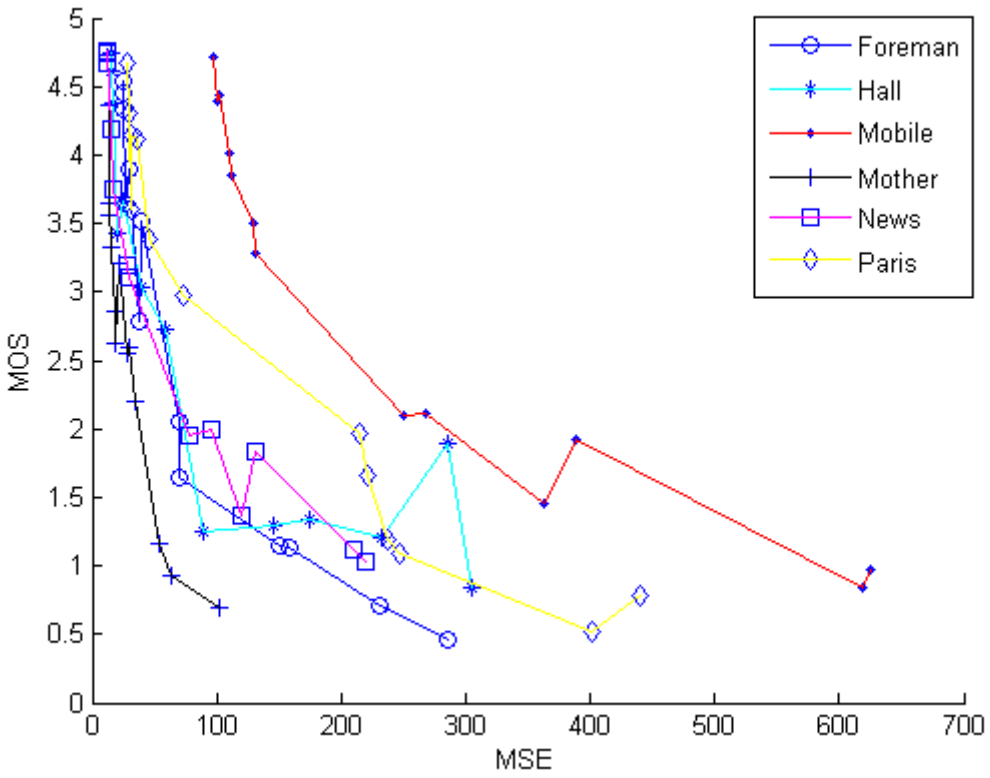


Figure 33 - MOS versus MSE for the PoliMi database

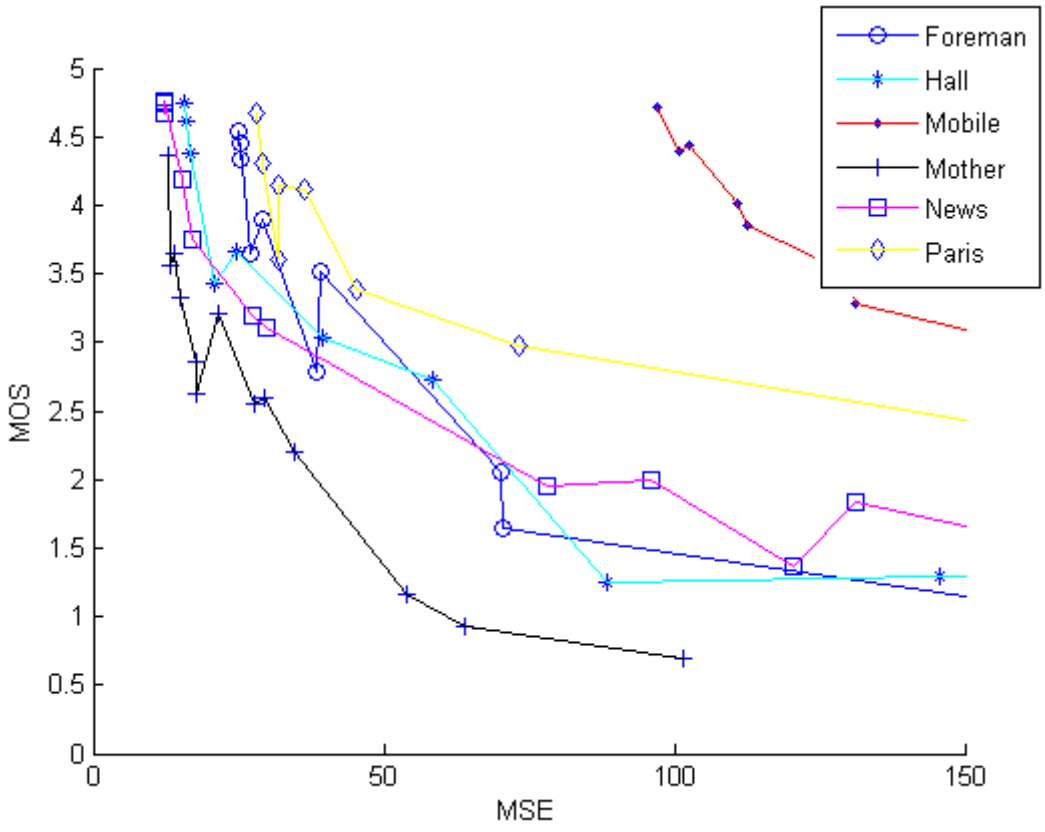


Figure 34 - MOS versus MSE for the PoliMi database and MSE values in the range 0 - 150

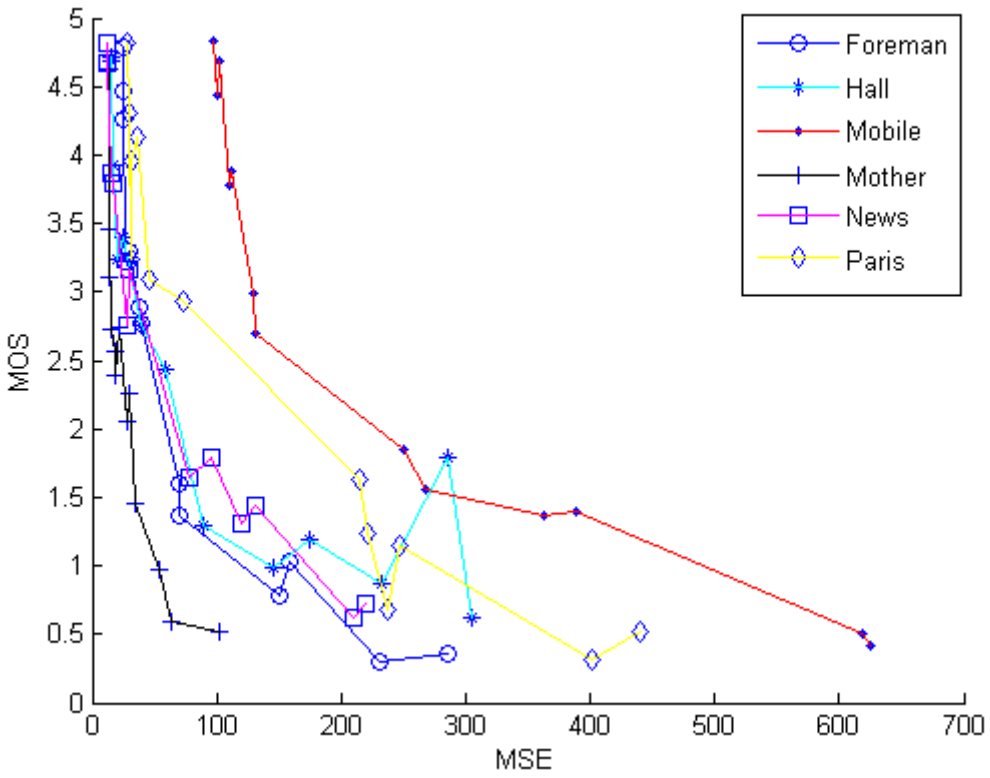


Figure 35 - MOS versus MSE for the EPFL database

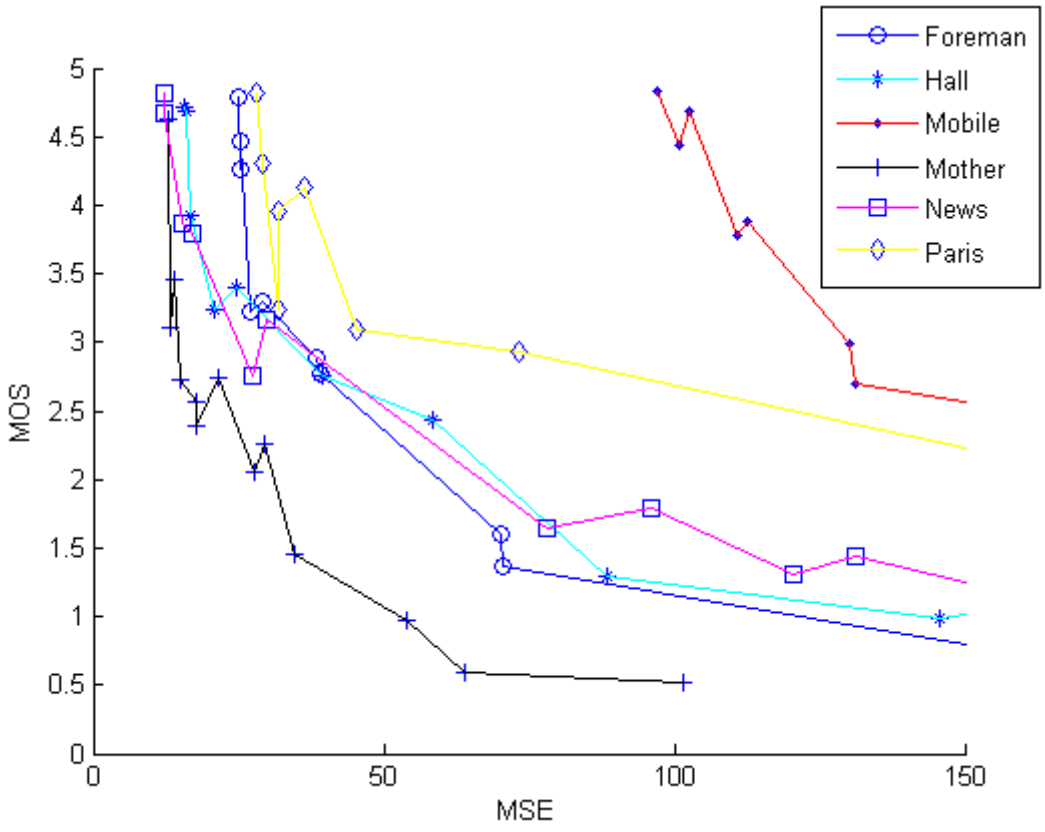


Figure 36 - MOS versus MSE for the EPFL database and MSE values in the range 0 - 150

If we look at each sequence individually, we can see that the MOS tends to decrease as the MSE increases. However, a closer look shows that, unlike what happens after compression, the plot MOS(MSE) does not have a monotonous variation, since there are situations where the MOS clearly increases when the MSE increases. For instance, for the sequence “Hall” in the EPFL database, an increase on the MSE from 232 to 285 resulted in an increase of the MOS from 0.8 to 1.8. With these observations, a video quality prediction model based on the MSE seems to be potentially unreliable for quality prediction of videos affected by packet losses. Thus a different approach is necessary.

Since the new element introduced were the packet losses, characterized by the PLR, the relation between PLR and MOS was analyzed. We started by computing the actual PLR of each video sequence by analysing the bitstream and checking the syntax of the packet header on each transmitted packet. Figure 37 and Figure 38 present the resulting MOS values *vs* PLR for each database.

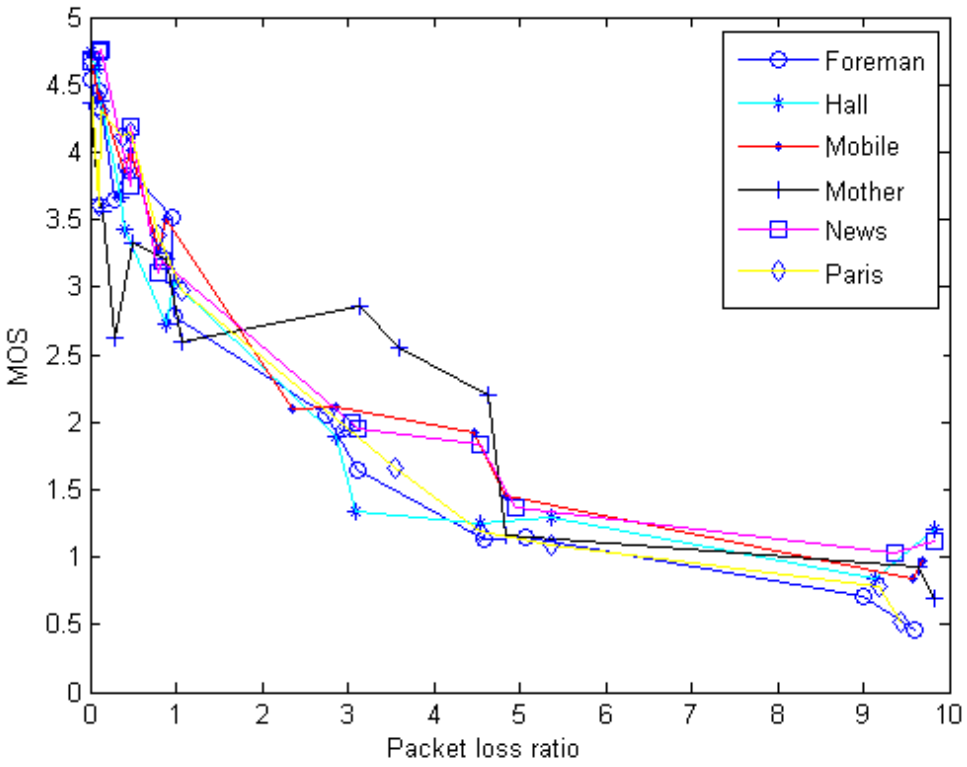


Figure 37 - MOS versus PLR for the PoliMi database

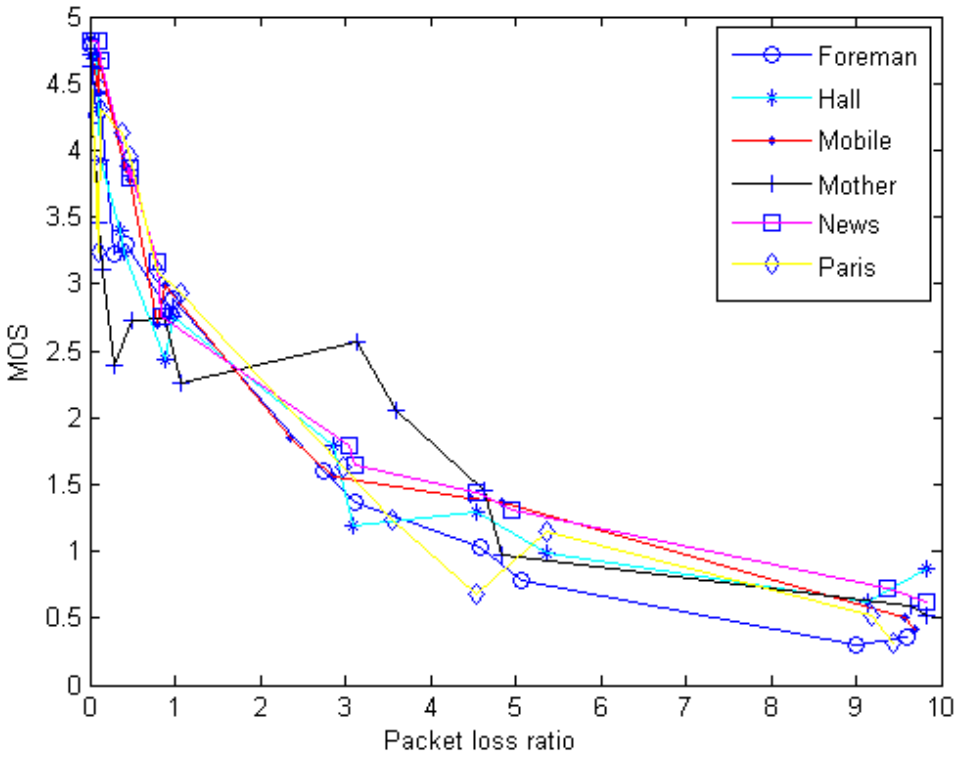


Figure 38 - MOS versus PLR for the EPFL database

The MOS versus PLR suggest that MOS values are better correlated with PLR than with MSE. To further confirm this, the Spearman correlation metric between MOS and PLR or MSE were calculated and are presented in Table 12.

Table 12 - Spearman metric for MOS/MSE and MOS/PLR

Correlation metric	**MOS versus MSE**	**MOS versus PLR**
Spearman	-0.7511	-0.9518

The values of the Spearman metric confirm that the MOS has a better correlation with the PLR. Taking this into consideration, all the following prediction models are based on the MOS/PLR relationship.

4.4 Simple PLR model

4.4.1 Model description

This model is based on the video quality prediction model proposed in ITU-T Rec. G.1070

[ITUT07] and it relates the MOS with the PLR. Figure 37 and Figure 38 suggest that the MOS/PLR relation can be described by an exponential function, thus the model is mathematically given by:

$$\mathrm{MOS} = MOS_{pl0} \times \exp\left(-\frac{PLR}{\theta}\right) \quad (18)$$

where MOS_{pl0} is the MOS of the video without any transmission losses, PLR is the packet loss ratio and θ is a parameter. The θ parameter of each video sequence can be obtained by regression using the MOS and the real PLR values. Table 13 shows the values obtained for each video sequence for both databases.

Table 13 – θ parameter value of each video sequence for both databases

Database	Foreman	Hall	Mobile	Mother	News	Paris
θ PoliMi	3.28	2.87	3.88	5.08	4.06	3.38
θ EPFL	2.19	2.33	2.56	3.10	3.00	2.54

After obtaining the value of θ for each sequence, the regression curves can be computed. Figure 39 shows the regression curve for "Paris" and the MOS data points that originated the curve.

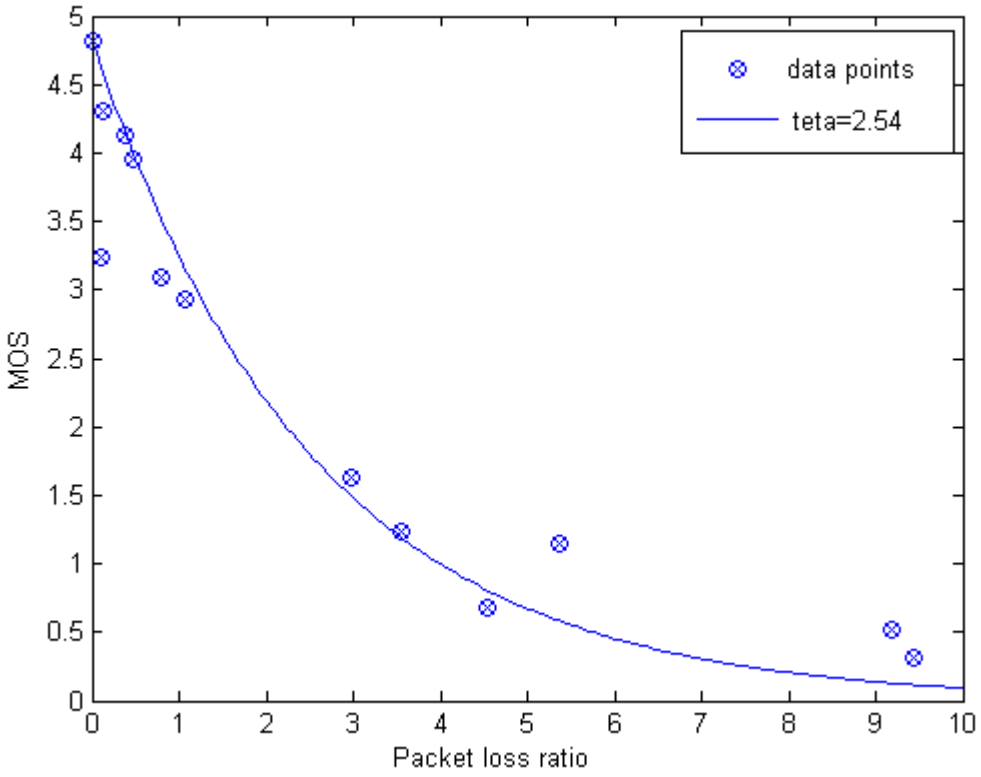


Figure 39 - Regression curves for "Paris" video sequence

Similarly to what was described in the previous chapter, an attempt to relate the exponential parameters, θ, with the video activities was made; however, as seen in Figure 40, a good correlation was not found.

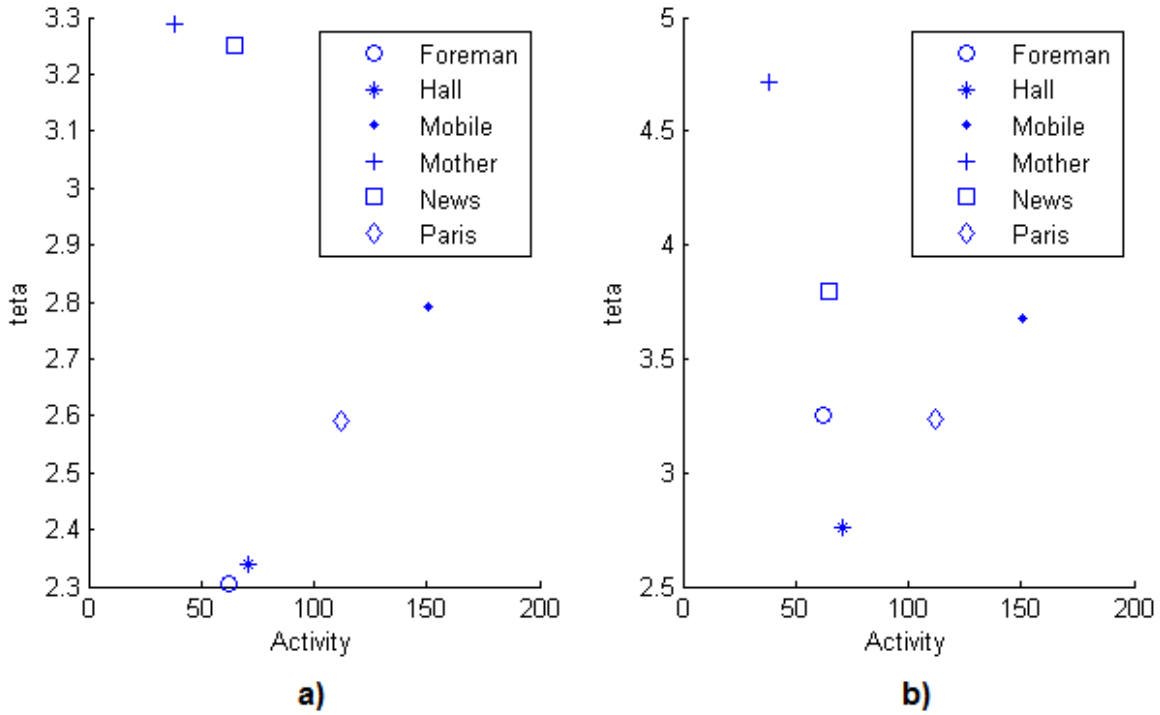


Figure 40 - Video activity versus θ for both databases a) EPFL b) PoliMi

The plots of the video sequences, as seen in Figure 37 and Figure 38, almost overlap each other and the θ values seem to have little variation from video to video. So it was decided to consider θ a constant. The value of this constant was obtained by regression through a procedure that will be explained in next section.

The MOS_{PL0} is the MOS when the video has no packet losses; the corresponding values are available from the subjective test results (FR model) or can be estimated (NR model) with the method described in the previous chapter.

4.4.2 Results and model validation

Since only six video sequences are available a cross-validation training method was used to train and test the model. Namely, the *leave-one-out cross-validation* was utilized. Figure 41 and Figure 42 plot the resulting MOSp versus subjective MOS values, for both databases using the FR model for computing MOS_{PL0}; Table 14 and Table 15 show the corresponding correlation metrics and RMS values.

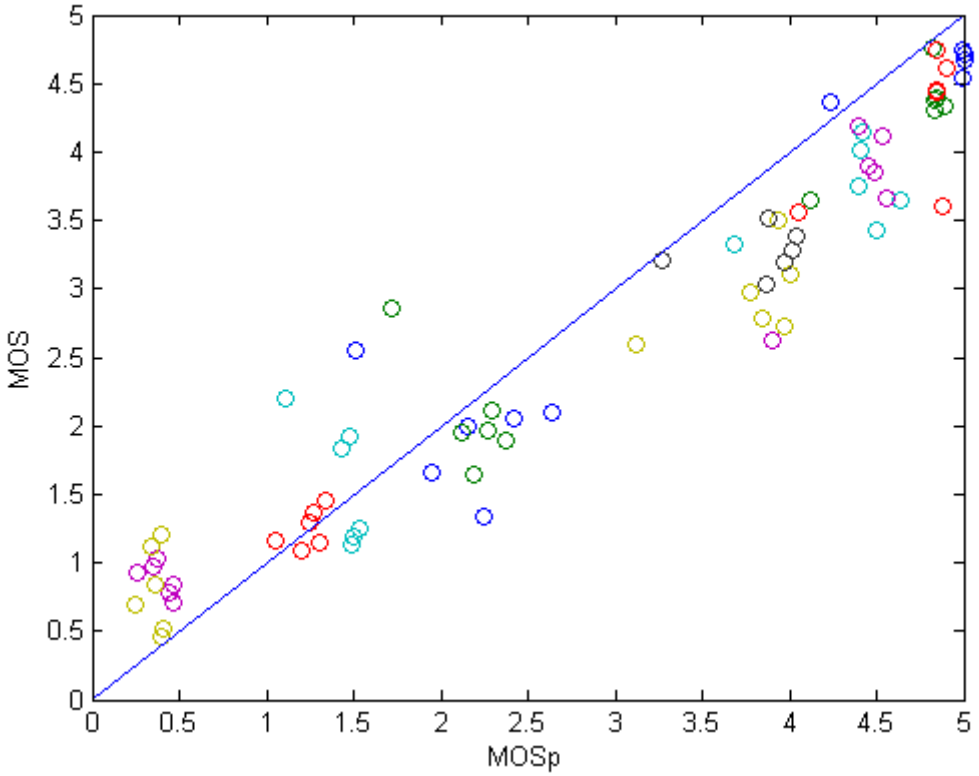


Figure 41 - MOS versus MOSp for the PoliMi database for the FR Simple Model

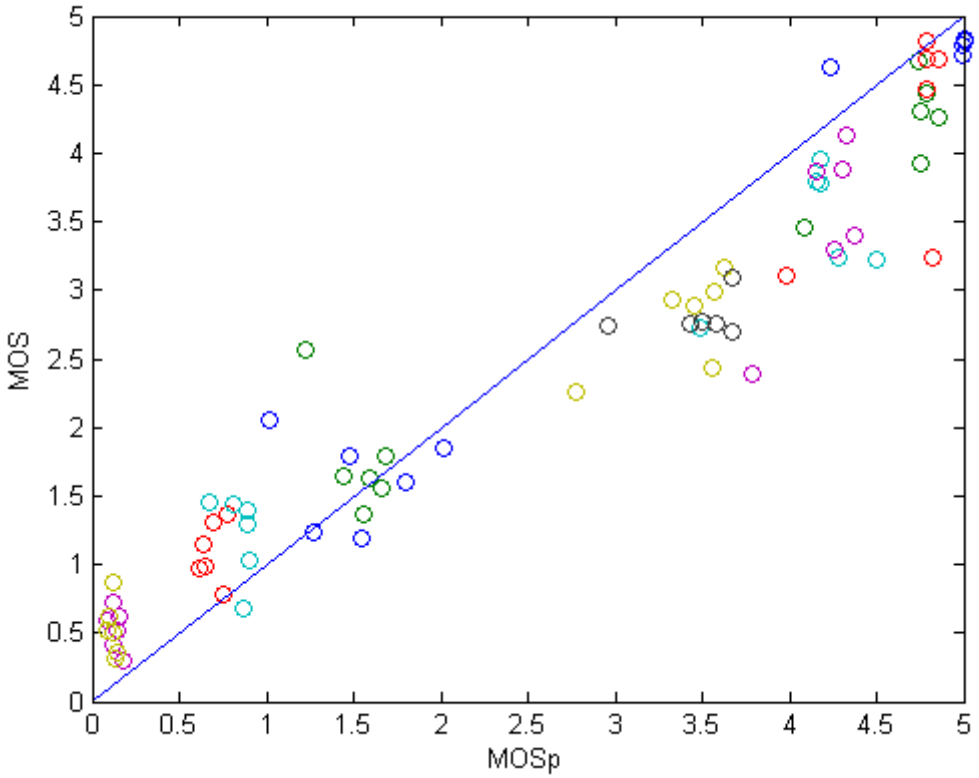


Figure 42 - MOS versus MOSp for the EPFL database for the FR Simple Model

Table 14 - Correlation metrics for individual video sequences using the FR Simple model

Correlation Coefficient	**Pearson**		**Spearman**		**RMS**	
Database	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*
Foreman	0.986	0.982	0.984	0.978	0.299	0.453
Hall	0.951	0.965	0.984	0.984	0.577	0.541
Mobile	0.986	0.981	0.977	0.993	0.354	0.393
Mother	0.890	0.877	0.956	0.951	0.747	0.923
News	0.972	0.983	0.966	0.999	0.421	0.429
Paris	0.983	0.973	0.962	0.962	0.361	0.468

Table 15 - Correlation metrics using the FR Simple model for all video sequences

Database	**Pearson**	**Spearman**	**RMS**
PoliMi	0.957	0.959	0.485
EPFL	0.956	0.959	0.564

The model was also evaluated as a NR model by using an estimation of the MOS_{PL0}. Figure 43 and Figure 44 plot the obtained MOSp versus the subjective MOS values, for both databases, using the NR model; Table 16 and Table 17 show the correlation metrics and RMS values obtained.

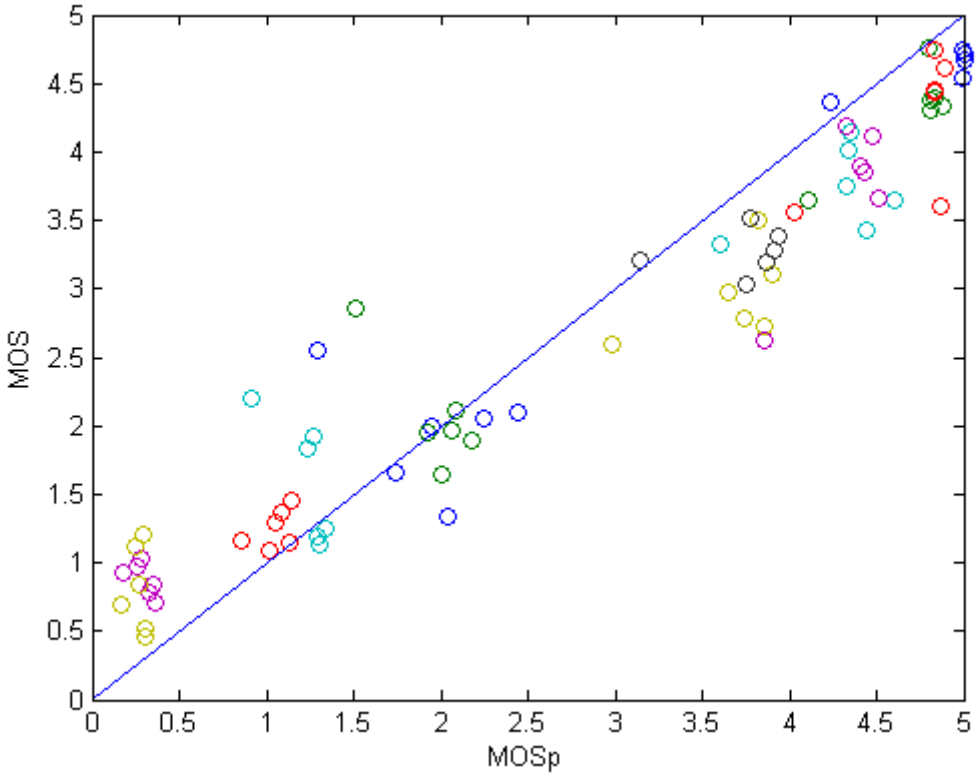


Figure 43 - MOS versus MOSp for the PoliMi database for the NR Simple Model

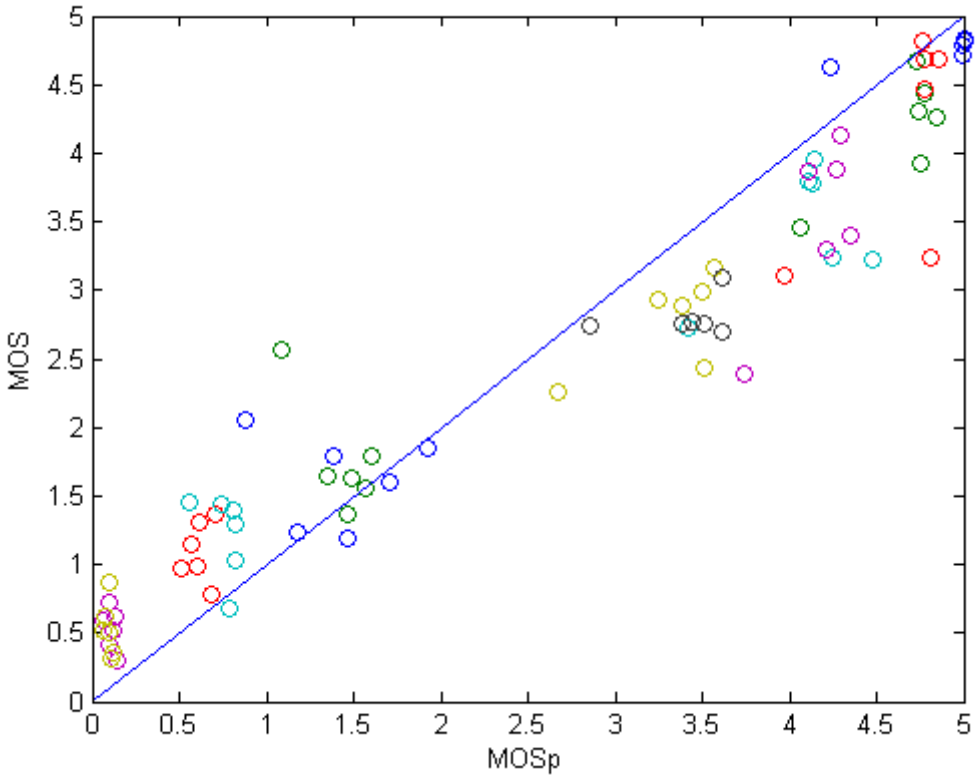


Figure 44 - MOS versus MOSp for the EPFL database for the NR Simple Model

Table 16 - Correlation metrics for individual video sequences using the NR Simple model

Correlation Coefficient	**Pearson**		**Spearman**		**RMS**	
Database	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*
Foreman	0.989	0.983	0.983	0.978	0.490	0.535
Hall	0.958	0.967	0.983	0.984	0.658	0.653
Mobile	0.989	0.982	0.977	0.993	0.467	0.444
Mother	0.883	0.874	0.956	0.951	0.795	0.826
News	0.976	0.984	0.966	0.999	0.507	0.468
Paris	0.984	0.973	0.962	0.962	0.495	0.534

Table 17 - Correlation metrics using the NR Simple model for all video sequences

Database	**Pearson**	**Spearman**	**RMS**
PoliMi	0.959	0.956	0.581
EPFL	0.960	0.963	0.591

The results show that the Simple PLR FR model scored acceptable values for the Pearson, Spearman and RMS metrics. Although the introduction of the estimated MOS_{PL0} caused small variations in the correlation metrics, the Simple PLR NR model was still able to predict acceptable values for the MOS.

Despite these positive results, there are a few particular cases where the Simple PLR model didn't have a good performance. An example is the sequence "Mother" where the model scored a Pearson value of 0.88 for PoliMi and 0.87 for EPFL; also, for low PLR values (*i.e.*, PLR < 1%), there are some cases in which an increase in PLR is accompanied by an increase in MOS. In the next sections we will address possible modifications to this simple model.

4.4.3 Motivation

Figure 37 and Figure 38 suggest that the MOS is related with the PLR by an exponential function. However, there are cases where the PLR increases but the MOS also increases, when it was

expected to decrease. There are also cases where a small increase in the PLR results in a high decrease of the MOS. Both of these cases can be observed in Figure 45, where the MOS versus PLR plot is shown for the “Mother” sequence and for the PoliMi database.

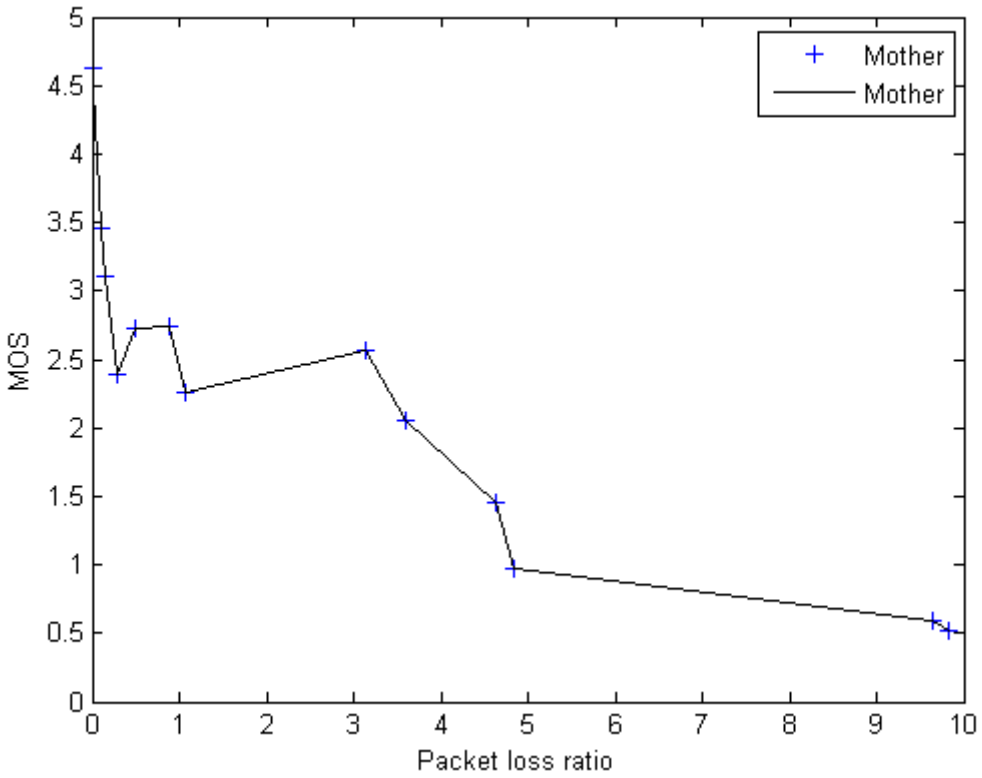


Figure 45 - MOS versus PLR for the sequence "Mother"

When observing the plot of MOS versus PLR, these situations usually result in the data points which are further away from the expected exponential line. This can happen due to various reasons, such as:

- The different subjective impact caused by errors when they occur in I, P or B-frames.
- Error propagation due to the use of P and B frames.
- The different subjective impact caused by errors when they occur in zones with high or low temporal activity.
- The different subjective impact caused by errors when they occur in zones with high or low spatial activity.
- The error pattern.

The models described in this section were developed seeking better adaptation to these particular situations.

4.4.4 Effective PLR

During H.264/AVC encoding I, P and B-frames are used, creating some dependencies between frames. As a consequence, a packet loss can affect the quality of, not only the frame where it happens, but also any frame that depends on the frame where the loss occurred. Figure 46 shows a frame where no packet losses have occurred; however, errors are visible due to error propagation. The structure of the frame dependency can be known by using the MVs that are conveyed in the encoded video bitstream. The video sequences utilized were encoded using the High Profile with each 4x4 block from a P or B-frame having one or two MVs, respectively.

Figure 46 - Additional errors due to error propagation

It should be noted that for the test video sequences, a *slice* has a dimension of 16x352 pixels. This means that a *slice* has 4x88 blocks of 4x4 pixels and that an additional lost block adds

$$\frac{1}{n\ block\ per\ slice \times n\ slices\ per\ frame \times n\ frames} = \frac{1}{352 \times 18 \times 298} \approx 5.2963 \times 10^{-5}\%$$

to the PLR, where *n blocks per slice* is the number of blocks per *slice*, *n slices per frame* is the number of slices per frame and *n frames* is the total number of frames.

Using the additional information given by the MVs it can be determined the effective packet loss ratio which is the real PLR plus the additional PLR due to frame dependency and error propagation. Figure 47 plots the effective packet loss ratio versus the MOS for the PoliMi database.

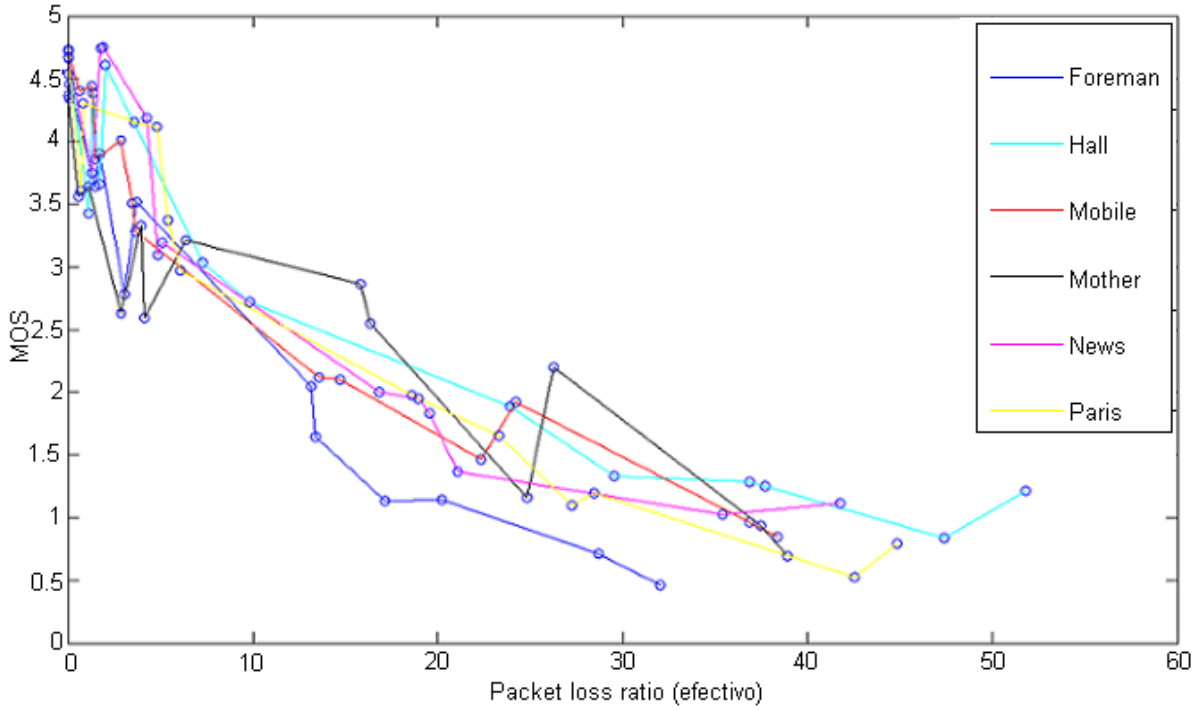


Figure 47 - MOS versus effective PLR for the PoliMi database

By observing Figure 47 it can be seen that the initial situations which motivated theses prediction models are still there. This can be justified by the fact that the error concealment technique, used by the decoder, is sometimes able to conceal the errors. Figure 48 shows an example of successful error concealment.

Figure 48 - Successful error concealment a) where the loss occurred b) decoder output

This prevents some lost blocks from negatively affect the video quality. Also, a loss in an I-frame is expected to be more relevant that a loss in a B-frame. Therefore, the effective PLR is right at considering the dependencies but wrong at not considering the error concealment techniques used by the decoder.

4.4.5 Frame Type Model

Since the H.264 uses frame dependency, it is expected that the degradation caused by a packet loss on a video sequence will depend on the type of frame where the loss occurs. For the used encoding, a packet loss in an I-frame is expected to be more relevant than a loss in a B-frame since an I-frame has frames which depend on it, while a B-frame does not. Moreover, the decoder uses, as error concealment, intra-frame prediction for I-frames and inter-frame prediction for P and B frames. This difference in the concealment technique reinforces the idea that losses should be discriminated by the frame type where they occur.

The model described and analyzed in this section, separates the packet losses according to the type of frame where they occur, giving them different weights. It tries to improve the simple PLR model by using a modified PLR

$$\mathrm{MOSp} = MOS_{pl0} \times exp(-fPL) \tag{19}$$

where,

$$fPL = \left(\frac{\omega_I.\sum I\ Block\ loss + \omega_P.\sum P\ Block\ loss + \omega_B.\sum B\ Block\ loss}{\sum total\ blocks}\right) \tag{20}$$

being, *fPL* the modified PLR, MOS_{pl0} the MOS of the video sequence without any transmission losses, ω_j the weight of the j-type frames, $\sum$ *j Block loss* the total of lost 4x4 blocks belonging to a *j*-type frame and $\sum$ *total blocks* the total number of 4x4 blocks in the video.

Figure 49 and Figure 50 represent the resulting *fPL* values versus the MOS for both databases.

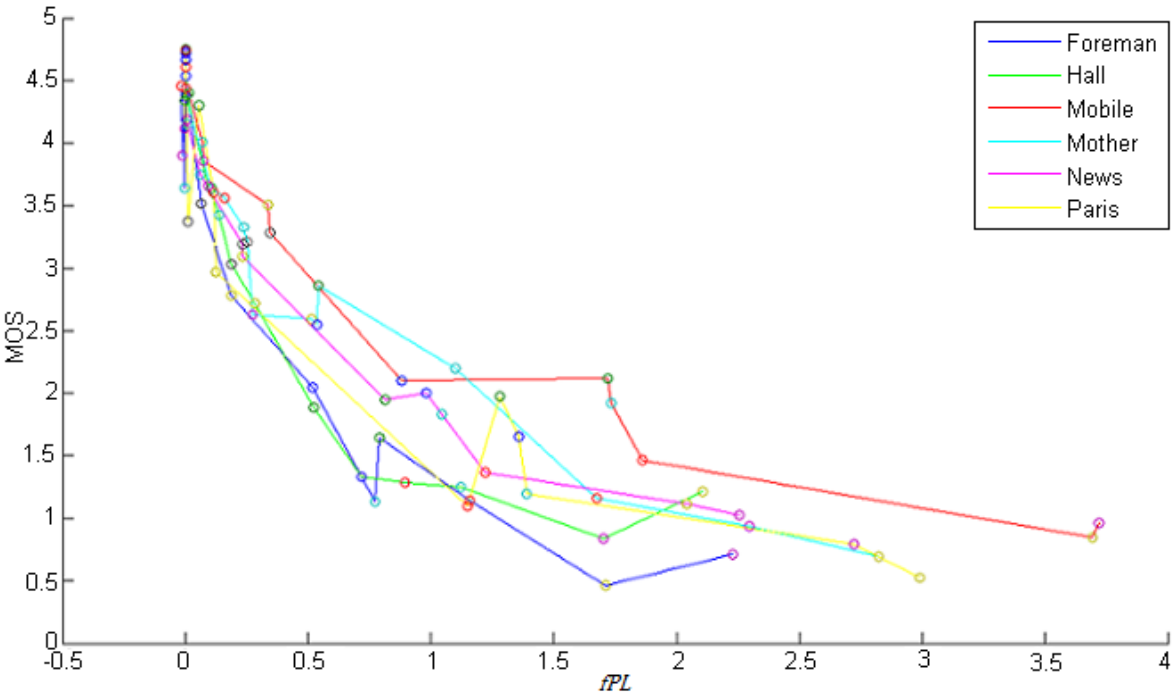


Figure 49 - MOS versus modified PLR for the PoliMi database

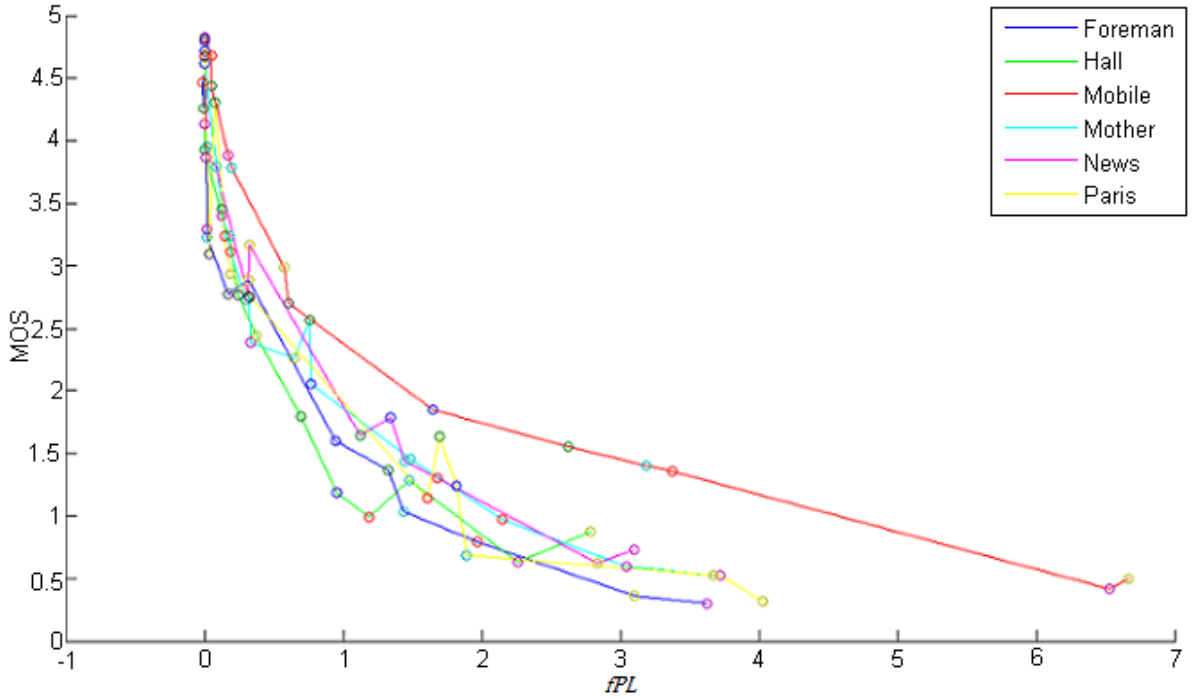


Figure 50 - MOS versus modified PLR for the EPFL database

Besides the exponential relation between *fPL* and MOS, Figure 49 and Figure 50 show a more monotonically relation between the two when compared to the MOS and PLR relation. This difference is particularly notorious for the sequence “Mother”, which translates into a better Pearson value for this sequence. In order to validate the model, the *leave-one-out cross-validation* method was used. Figure 51 and Figure 52 show the MOS versus MOSp for the Frame Type model while Table 18 and Table 19 show the resulting correlation metrics.

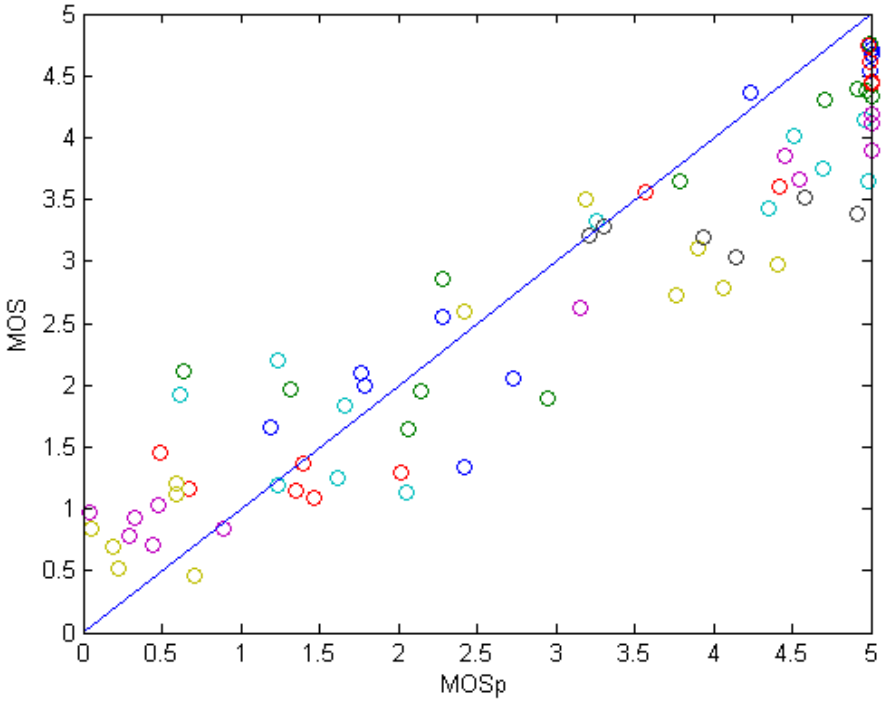


Figure 51 - MOS versus MOSp for the PoliMi database for the Frame Type Model

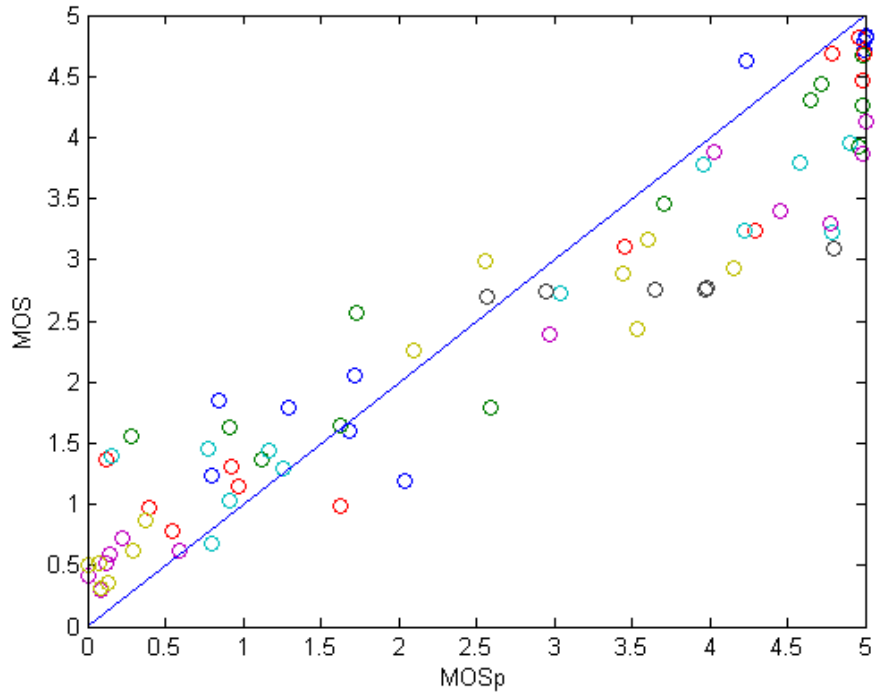


Figure 52 - MOS versus MOSp for the EPFL database for the Frame Type Model

Table 18 - Correlation metrics for individual video sequences using the Frame Type Model

Correlation Coefficient	**Pearson**		**Spearman**		**RMS**	
Database	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*
Foreman	0.974	0.968	0.950	0.984	0.802	0.758
Hall	0.949	0.957	0.995	0.978	0.777	0.782
Mobile	0.982	0.982	0.982	0.995	0.775	0.711
Mother	0.971	0.957	0.984	0.962	0.445	0.468
News	0.979	0.978	0.949	0.962	0.532	0.543
Paris	0.961	0.961	0.905	0.935	0.779	0.796

Table 19 - Correlation metrics using the Frame Type Model for all video sequences

Database	**Pearson**	**Spearman**	**RMS**
PoliMi	0.941	0.935	0.699
EPFL	0.949	0.952	0.688

When compared with the simple PLR model, the frame type model scored slightly worse results in all three correlation coefficients. A reason for this may be the fact that this model only considers the frame type where the losses occur and ignores the subjective impact cause by error propagation (due to frame dependency) or spatial-temporal activity.

4.4.6 Frame Type and Movement Model

As previously mentioned, video decoders use error concealment techniques to try to prevent video degradation caused by packet losses. Some techniques work better than others however, all of them can more efficiently conceal a loss when the video sequence doesn't have much movement. The model described and analyzed in this section, adds this information to the frame type model of section 4.4.5. This is done by only considering a lost 4x4 block in a P or B frame as an actual loss, if the norm of its motion vector ($MVabs$) is higher than a threshold value. Losses occurring in an I-frame are always considered as actual losses.

The *norm of a motion vector* is computed by:

$$MVabs = \sqrt{MVx^2 + MVy^2} \tag{21}$$

where MVx and MVy are, respectively, the x-axis component and the y-axis component of the motion vector.

Mathematically, the model is given by:

$$\text{MOSp} = MOS_{pl0} \times \exp\left(fPL_{mv}\right) \tag{22}$$

where,

$$fPL_{mv} = \frac{\omega_I.\sum I\ Block\ loss + \omega_P.\sum P\ Block\ loss\ \omega_B.\sum B\ Block\ loss}{\sum total\ blocks} \tag{23}$$

being fPL_{mv} the modified PLR, MOS_{pl0} the MOS of the video sequence without any transmission losses, ω_j the weight of the *j*-type frames, $\sum j\ Block\ loss$ the total of actual lost 4x4 blocks belonging to a *j*-type frame and $\sum total\ blocks$ the total number of 4x4 blocks in the video.

To choose the value of the threshold, the model was tested with various values and a threshold of 10 was the one producing the better results. It should be noted that the MVabs were computed using the MVs associated to the lost blocks.

Once again, the *leave-one-out cross-validation* method was used to validate the model. Figure 53 and Figure 54 show the MOS versus the obtained MOSp, while Table 20 and Table 21 show the resulting correlation metrics.

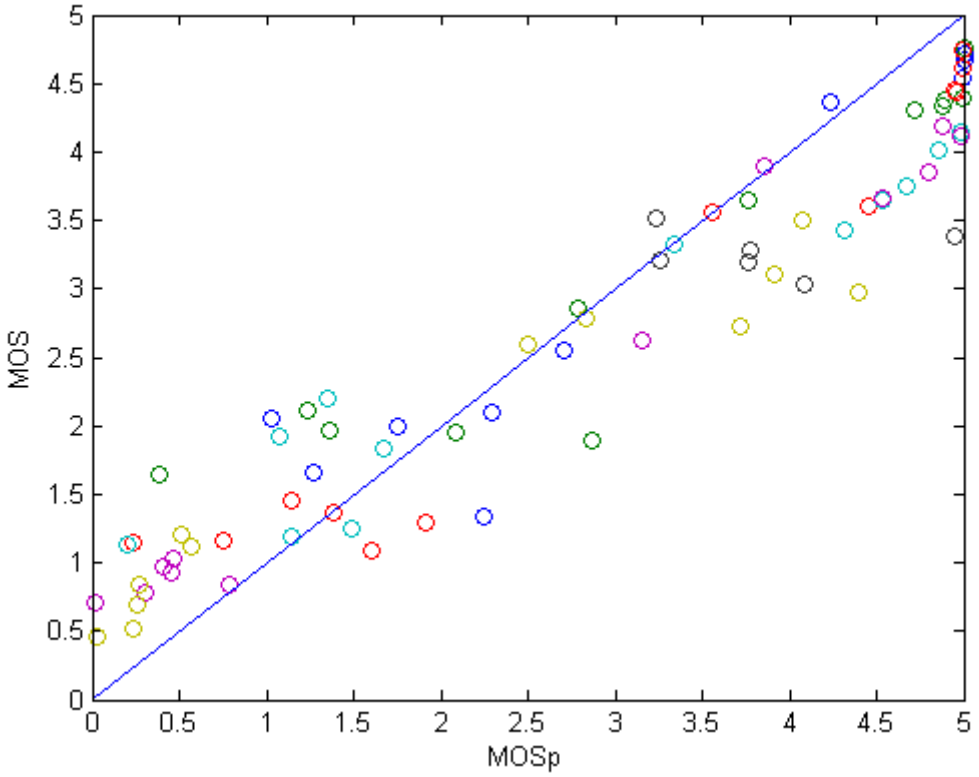


Figure 53 - MOS versus MOSp for the PoliMi database for the Frame Type and Movement Model

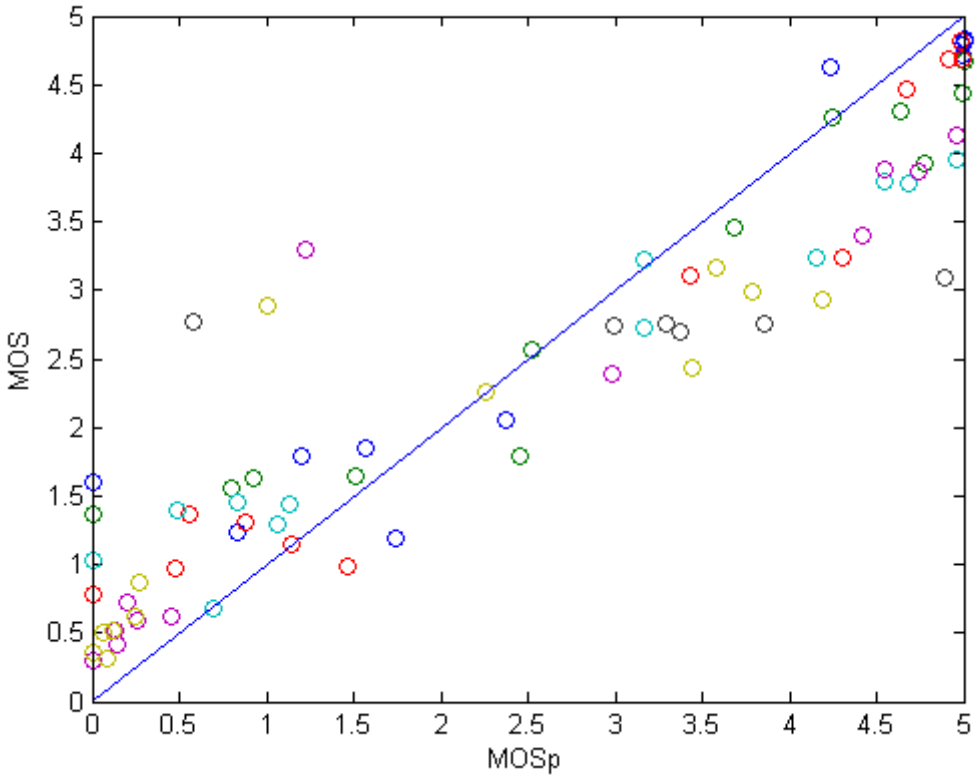


Figure 54 - MOS versus MOSp for the EPFL database for the Frame Type and Movement Model

Table 20 - Correlation metrics for individual video sequences using the Frame Type and Movement Model

Correlation Coefficient	**Pearson**		**Spearman**		**RMS**	
Database	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*
Foreman	0.980	0.902	0.989	0.973	0.714	1.211
Hall	0.953	0.963	0.995	0.978	0.726	0.700
Mobile	0.981	0.976	0.984	0.978	0.630	0.627
Mother	0.974	0.965	0.989	0.984	0.359	0.383
News	0.981	0.986	0.979	0.979	0.503	0.481
Paris	0.959	0.959	0.923	0.934	0.786	0.815

Table 21 - Correlation metrics using the Frame Type and Movement Model for all video sequences

Database	**Pearson**	**Spearman**	**RMS**
PoliMi	0.958	0.956	0.637
EPFL	0.933	0.933	0.752

The results show that the Frame Type and Movement Model has scored acceptable values for the correlation coefficients. However, when compared with the Simple PLR Model, this model scored worse results, especially for the RMS metric.

4.4.7 Frame Type, Dependencies and Movement Model

This model takes into account the frame type where the losses occur, the additional losses as a result of the dependency between I, P and B-frames and the movement in the area where the losses occurred. Once again, a 4x4 block is only considered as an actual loss if its $MVabs$ is higher than a threshold (which assumes that the concealment technique used by the decoder is able to properly conceal a loss in a low movement area). This is also done to the additional losses resulting from error

propagation.

This model is mathematically given by:

$$\text{MOSp} = MOS_{pl0} \times exp(-fPL2) \tag{24}$$

where,

$$fPL2 = \frac{\omega_I.\sum I\ Blk\ loss + \omega_P.\sum P\ Blk\ loss + \omega_B.\sum B\ Blk\ loss + \omega_{DI}.\sum Dep\ I\ Blk\ loss + \omega_{DP}.\sum Dep\ P\ Blk\ loss}{\sum total\ blocks} \tag{25}$$

being $fPL2$ the modified PLR considering frame dependency, MOS_{pl0} the MOS of the video sequence without any transmission losses, ω_j the weight of the j-type frames, $\sum j\ Block\ loss$ the total of actual lost 4x4 blocks belonging to a *j*-type frame, $\sum Dep\ j\ Blk\ loss$ the total of 4x4 blocks received and with a $MVabs$ higher than the threshold (but dependent on lost 4x4 blocks belonging to a *j*-type frame) and $\sum total\ blocks$ the total number of 4x4 blocks in the video.

To choose the value of the threshold, the model was tested with various values and a threshold of 25 was the one producing the better results. It should be once again noted that the $MVabs$ were calculated with the MVs of the lost blocks.

Once again the *leave-one-out cross-validation* method was used to validate the model. Figure 55 and Figure 56 show the MOS versus the obtained MOSp while Table 22 and Table 23 show the resulting correlation metrics.

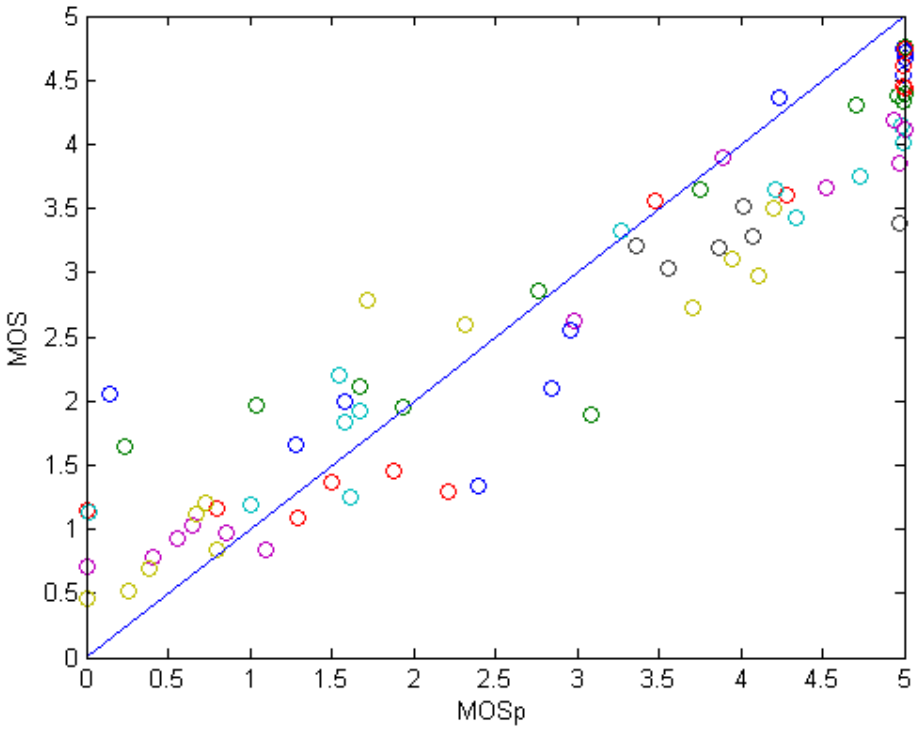


Figure 55 - MOS versus MOSp for the PoliMi database for the Frame Type, Dependencies and Movement Model

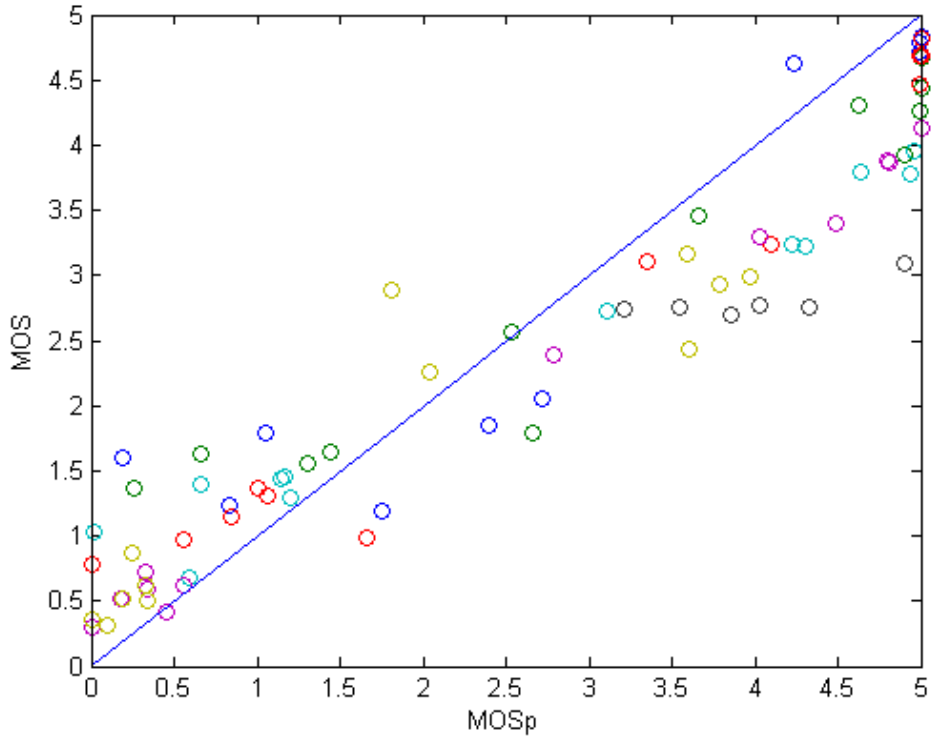


Figure 56 - MOS versus MOSp for the EPFL database for the Frame Type, Dependencies and Movement Model

Table 22 - Performance metrics for individual video sequences using the Frame Type, Dependencies and Movement Model

	Pearson		**Spearman**		**RMS**	
Database	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*
Foreman	0.909	0.869	0.949	0.922	1.222	1.313
Hall	0.960	0.970	0.995	0.978	0.672	0.638
Mobile	0.974	0.970	0.956	0.961	0.648	0.670
Mother	0.980	0.972	0.989	0.984	0.340	0.342
News	0.980	0.979	0.989	0.978	0.528	0.550
Paris	0.959	0.956	0.911	0.935	0.795	0.843

Table 23 - Performance metrics using the Frame Type, Dependencies and Movement Model for all video sequences

Database	Pearson	Spearman	RMS
PoliMi	0.947	0.945	0.678
EPFL	0.952	0.955	0.703

The results show that the Frame Type, Dependencies and Movement Model has scored acceptable values for the correlation coefficients, although sequence “Foreman” scored poorly on the RMS metrics. When compared with the Simple PLR model, this model is worse in all the correlation metrics and it is also more complex.

4.5 Statistical Model

4.5.1 Motivation

The modified PLR models weren’t fully able to address the situation they were initially trying to solve. Characteristics such as the frame type where the losses occur are relevant, but there is another characteristic that affects a video’s perceived quality, the packet loss pattern. For instance, if packet losses are concentrated in a single frame they will have a higher impact on quality than if they were distributed between various frames. By analyzing the syntax of the packet headers on each transmitted packet, this pattern can be obtained and various statistical metrics of the losses distribution can be computed. The ones that prove to be helpful in predicting a video’s perceived quality are selected to be part of the Statistical Model. Then, the model is evaluated using the correlation metrics recommended by VQEG.

4.5.2 Statistical metrics

By analyzing the packet loss pattern the following statistical metrics, related with the losses distribution, were computed:

- Packet loss ratio (PLR).
- Maximum number of lost packets on the same I or P-frame.
- Average number of lost packets on I, P or all frames with more than one loss.
- Maximum number of consecutive lost packets on the same I or P-frame.

- Average number of consecutive lost packets per I or P-frame, considering or ignoring *single losses* (a loss is a *single loss* if the previous and following packets were well received).
- Average distance (in *slices*) between lost slices and the center of the frame, per I, P or all frame.
- The Modified PLR from the Frame type model (*fPL*), proposed in section 4.4.5.

The correlation between each metric and the MOS was then determined, using the Spearman correlation metric; the resulting correlation values are presented in Table 24.

Table 24 - Spearman coefficient for each statistical metric

Statistical metric	Spearman
Packet loss ratio	-0,953
Maximum number of lost packets on the same I-frame	-0,909
Maximum number of lost packets on the same P-frame	-0,832
Average number of lost packets on frames with more than one loss	-0,841
Average number of lost packets on I-frames with more than one loss	-0,819
Average number of lost packets on P-frames with more than one loss	-0,659
Maximum number of consecutive lost packets on the same I-frame	-0,903
Maximum number of consecutive lost packets on the same P-frame	-0,829
Average number of consecutive lost packets per I-frame	-0,959
Average number of consecutive lost packets per I-frame, considering single losses	-0,824
Average number of consecutive lost packets per I-frame, ignoring single losses	-0,783
Average number of consecutive lost packets per P-frame	-0,922
Average number of consecutive lost packets per P-frame, considering single losses	-0,429
Average number of consecutive lost packets per P-frame, ignoring single losses	-0,477
Average distance between frames with losses, considering single losses	0,556
Average distance between frames with losses, ignoring single losses	0,377
Modified PLR from the Frame type model (PoliMi / EPFL)	-0.941 / -0.951

Some metrics are in fact well correlated with MOS values. A curious observation is that the "Average number of consecutive lost packets per I-frame" is slightly better correlated with MOS than the PLR and the modified PLR from the frame type model.

4.5.3 Selection of the statistical metrics

To select the appropriate model it is necessary to determine which variables should be used in the model. At the end, it is expected a model with enough variables so that it can perform satisfactorily; however, too many variables can overcomplicate the model. The variable selection was based on a *stepwise regression* [MoRu03].

4.5.3.1 *Stepwise regression*

Stepwise regression is one of the most used variable selection method. It can be used to differentiate the variables that should be included in the model from the ones that should be discarded. The method iteratively finds the regression model by adding or removing variables at each step, through a sequence of *f-tests*.

First a model is made with the variable which better correlates with the desired model output. Then a second model is made by adding the variable with the highest partial *f-statistic* to the first model. The *f-statistic* of this second variable is given by:

$$F_2 = \frac{\left(\frac{SSR_1 - SSR_2}{p2 - p1}\right)}{\left(\frac{SSR_2}{n - p2}\right)} \tag{26}$$

where SSR_i is the residual sum of squares of the *i-th* model, p_i is the number of parameters in the *i-th* model and *n* the number of data points to estimate the parameters of the models. The added variable is kept in the model only if its partial *f-statistic* is greater than the value of adding a variable to the model, f_{in}.

Suppose that $F_2 > f_{in}$ and that the second variable is kept in the model. Now the *stepwise regression* algorithm determines if the first added variable should be removed. This is done by calculating its partial *f-statistic*:

$$F_1 = \frac{\left(\frac{SSR_2 - SSR_1}{p1 - p2}\right)}{\left(\frac{SSR_1}{n - p1}\right)} \tag{27}$$

The variable stays in the model if its partial *f-statistic* is greater than the value of removing a variable from the model, f_{out}. The algorithm does the same with the remaining variables and stops when no variable can be removed or added to the model. Figure 57 shows a block diagram of the algorithm.

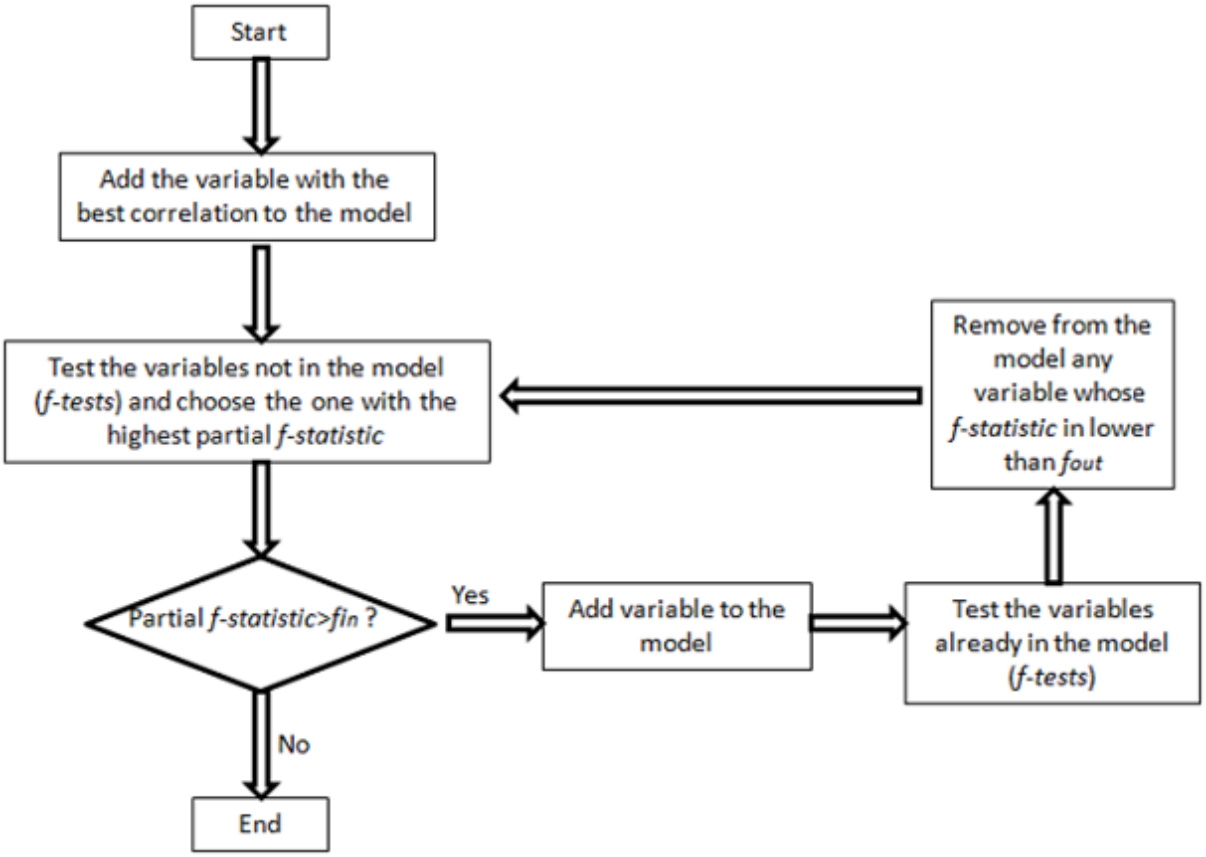


Figure 57 - Stepwise Regression

There are two simplifications of the stepwise *regression*, the forward selection and the backward elimination. The forward selection tests the variables one by one, adding them if their partial *f-statistic* is greater than f_{in}. However it does not test the variables already added to the model. The backward elimination starts with all variables in the model, then the variable with the smallest partial *f-statistic* is removed if it is lower than f_{out}. If the variable is removed, the process continues and the partial *f-statistics* of the remaining variables are recalculated to find the next variable for potential elimination. The process stops when no more variables can be removed. In this thesis, the algorithm was used without any of these simplifications.

4.5.3.2 *Applying the stepwise regression*

To select an appropriate value for f_{in} and f_{out} the *stepwise regression* was applied with various pairs of values. The pair of values selected was the one which produced the best final model. The first statistical metric to be used in the *stepwise regression* was the "average number of consecutive lost packets for all I-frames" since it has the best spearman correlation from all the statistical metrics (Table 24).

The best pair of values was f_{in}= 0.3500 and f_{out}= 0. With these values, the *stepwise regression* was applied (and using the PoliMi database) and the results are presented in Table 25.

Table 25 - Stepwise regression results

Statistical metric	In or Out
Packet loss ratio	Out
Maximum number of lost packets on the same I-frame	**In**
Maximum number of lost packets on the same P-frame	Out
Average number of lost packets on frames with more than one loss	**In**
Average number of lost packets on I-frames with more than one loss	Out
Average number of lost packets on P-frames with more than one loss	Out
Maximum number of consecutive lost packets on the same I-frame	Out
Maximum number of consecutive lost packets on the same P-frame	**In**
Average number of consecutive lost packets per I-frames	**In**
Average number of consecutive lost packets per I-frame, considering single losses	Out
Average number of consecutive lost packets per I-frame, ignoring single losses	Out
Average number of consecutive lost packets per P-frame	Out
Average number of consecutive lost packets per P-frame, considering single losses	Out
Average number of consecutive lost packets per P-frame, ignoring single losses	**In**
Average distance between frames with losses (considering single losses)	**In**
Average distance between frames with losses ignoring single losses	Out
Modified PLR from the Frame type model (PoliMi / EPFL)	**In**

So the final Statistical Model is mathematically given by:

$$\mathrm{MOSp} = MOS_{pl0} \times \exp\left(\sum_{i=1}^{n} \omega_i \times stat_i\right) \quad (28)$$

being n the number of statistical metrics (7 in this case), ω_i the weight of the i-th statistical metric and $stat_i$ the value of the i-th statistical metric.

4.5.4 Results and model validation

In order to validate the model the *leave-one-out cross-validation* method was used. In each turn, the weights are recalculated with the training set and the estimation of MOS values (MOSp) is obtained using the validation set. Figure 58 and Figure 59 depict the MOS versus the MOSp values, while Table 26 and Table 27 show the resulting correlation metrics.

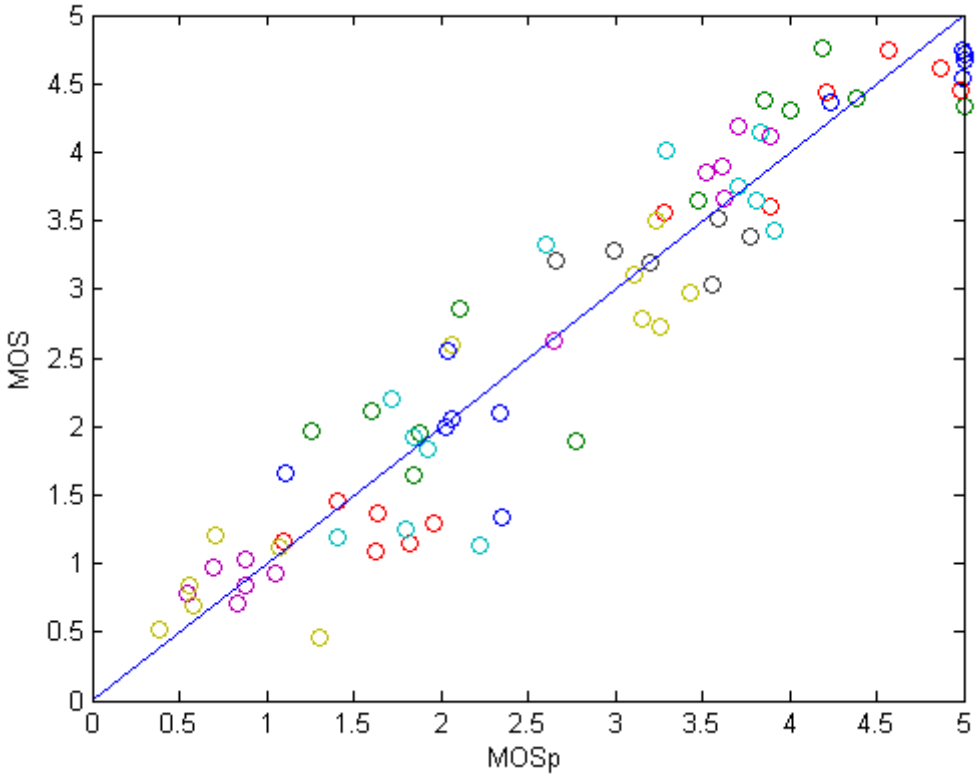


Figure 58 - MOS versus MOSp for the PoliMi database using the Statistical Model

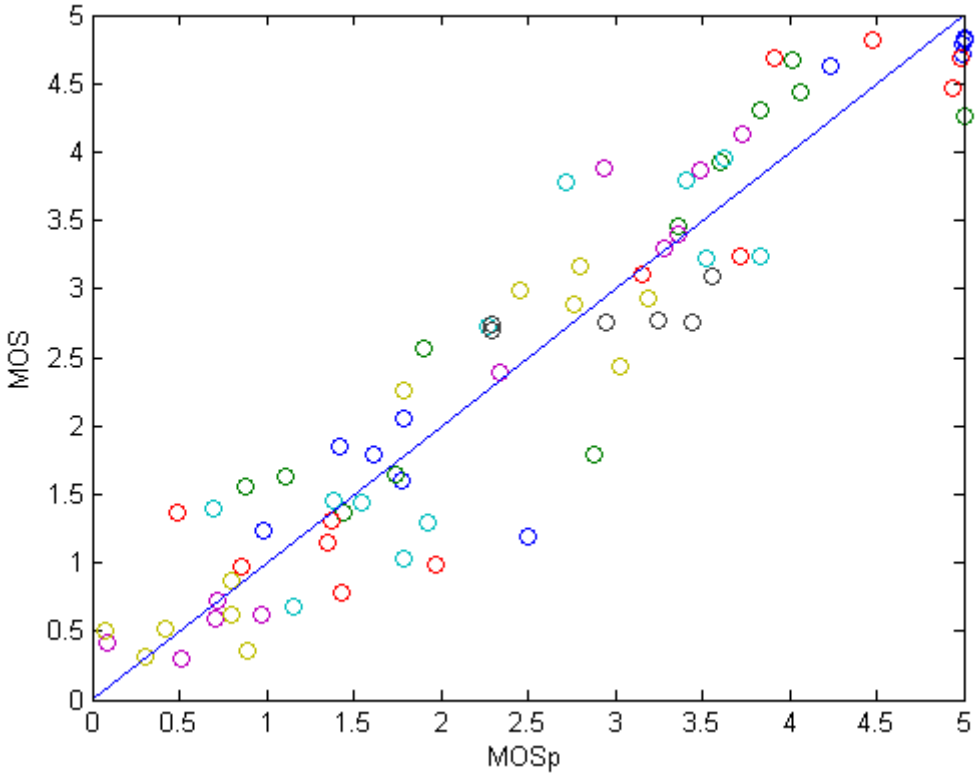


Figure 59 - MOS versus MOSp for the EPFL database using the Statistical Model

Table 26 - Performance metrics for individual video sequences using the Statistical Model

	Pearson		**Spearman**		**RMS**	
Database	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*	*PoliMi*	*EPFL*
Foreman	0.967	0.984	0.940	0.956	0.538	0.436
Hall	0.945	0.946	0.978	0.951	0.542	0.669
Mobile	0.980	0.980	0.973	0.989	0.302	0.646
Mother	0.966	0.980	0.973	0.962	0.418	0.323
News	0.985	0.987	0.967	0.984	0.251	0.295
Paris	0.965	0.973	0.934	0.951	0.399	0.356

Table 27 - Performance metrics using the Statistical Model, for all video sequences

Database	Pearson	Spearman	RMS
PoliMi	0.950	0.950	0.426
EPFL	0.945	0.942	0.479

The results show that the Statistical Model scored acceptable values for all three performance metrics. A comparison between this model and the previous ones is presented in the next section.

4.6 Results and model comparison

Figure 60 show the MOS vs. MOSp plots for the Simple PLR model and for the Statistical model. Here it can be seen that the Simple PLR model has a good performance, which translates into high correlation coefficient values, as shown in Table 28. However, a few predictions were far from the true values and that resulted in the development of the Modified PLR models. All the Modified PLR models have acceptable performances, as shown in Table 28, but were unable to significantly improve the Simple PLR model. In fact, their RMS values are higher than the Simple PLR model's RMS values.

The Statistical Model has a good performance (Pearson = 0.950 (PoliMi), 0.945 (EPFL); Spearman = 0.950 (PoliMi), 0.942 (EPFL)) when compared to the other models. For the Pearson and Spearman metrics, the model scored slightly lower than the Simple PLR model. But, as a plus, the model was able to address the situations where the Simple PLR model failed. This translates into better RMS values, since the Statistical Model obtained a RMS of 0.426 (PoliMi) and 0.479 (EPFL) while the Simple PLR model obtained a RMS of 0.581 (PoliMi) and 0.591 (EPFL). To further confirm this, Figure 61 and Figure 62 show the complementary cumulative distribution of the prediction errors for the Simple PLR model and for the Statistical Model using the PoliMi and the EPFL databases, respectively. We can observe that the number of predictions with a high prediction error is lower for the Statistical Model than for the Simple PLR model.

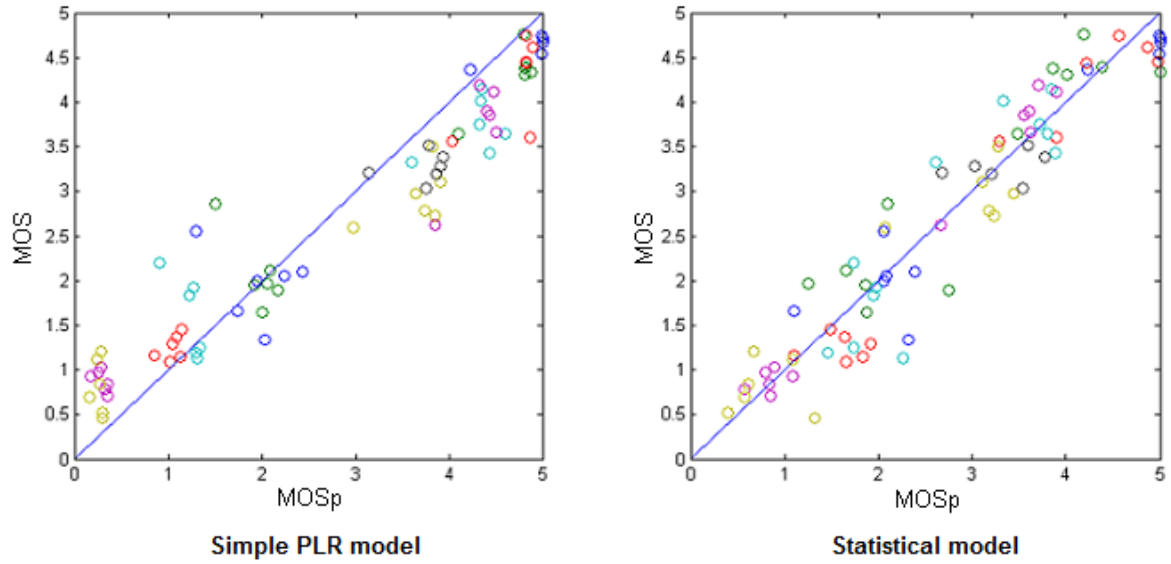


Figure 60 - MOS versus MOSp for the Simple PLR model and for the Statistical Model

Table 28 - Performance of each model

Model	Model performance			
Simple PLR Model	**Database**	**Pearson**	**Spearman**	**RMS**
	PoliMi	0.959	0.956	0.581
	EPFL	0.960	0.963	0.591
Frame Type Model	**Database**	**Pearson**	**Spearman**	**RMS**
	PoliMi	0.941	0.935	0.699
	EPFL	0.949	0.952	0.688
Frame Type and Movement Model Threshold 10	**Database**	**Pearson**	**Spearman**	**RMS**
	PoliMi	0.958	0.956	0.637
	EPFL	0.933	0.933	0.752
Frame Type, Dependencies and Movement Model Threshold 25	**Database**	**Pearson**	**Spearman**	**RMS**
	PoliMi	0.947	0.945	0.678
	EPFL	0.952	0.955	0.703
Statistical Model	**Database**	**Pearson**	**Spearman**	**RMS**
	PoliMi	0.950	0.950	0.426
	EPFL	0.945	0.942	0.479

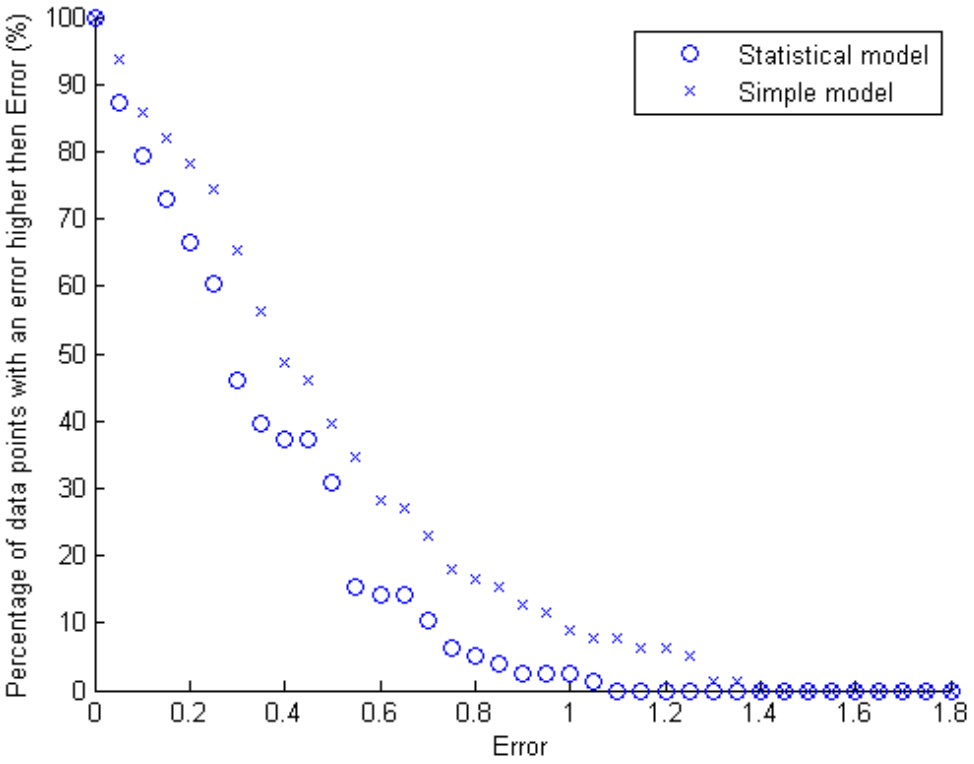


Figure 61 - Cumulative distribution function of the prediction errors (PoliMi database)

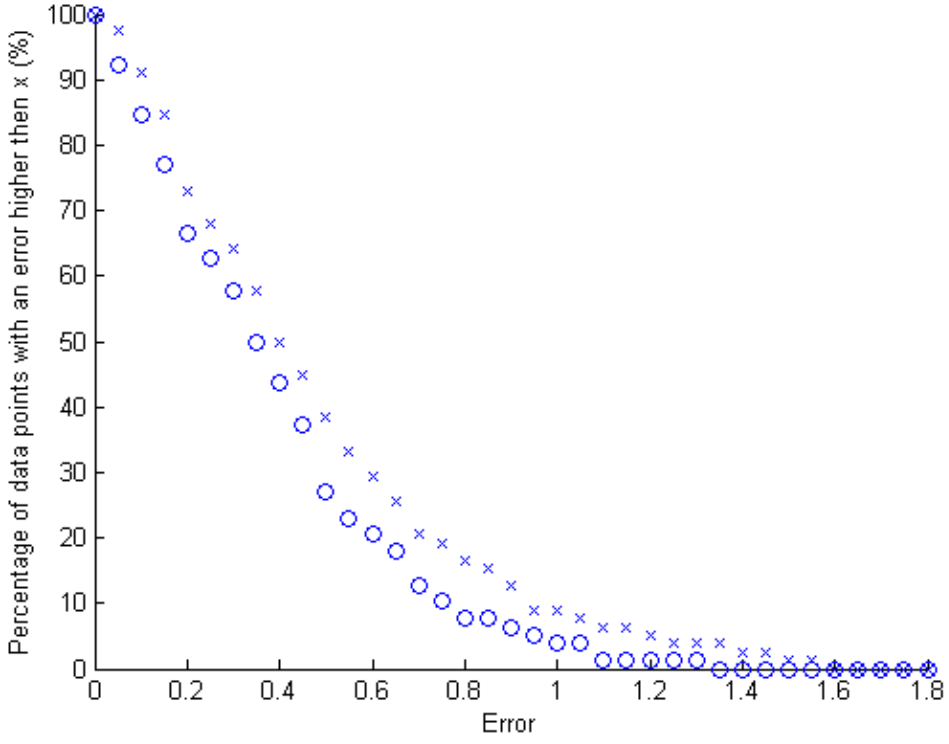


Figure 62 - Cumulative distribution function of the prediction errors (EPFL database)

4.7 Conclusion

In this chapter, objective video quality assessment in IP networks was addressed. First, a Simple PLR model, based on the ITU-T Rec. G.1070, was described and analysed. This analysis concluded that the Simple PLR model has a good performance, but is unable to predict the right MOS value for some particular cases.

Then, Modified PLR models were described and analysed. These models tried to improve the Simple PLR model by taking into consideration other factors, besides the PLR, that may have an impact on quality. The additional factors included the frame type where the losses occurred, frame dependencies, and frame movement. All the Modified PLR models have acceptable performance; however, they aren't as good as the Simple PLR model.

Afterwards, a statistical model was described and analysed. This model takes into consideration not only features such as the frame type but also the packet loss pattern. This pattern can be obtained by analysing the syntax of the packet headers and allows the computation of numerous statistical metrics that try to characterise the losses pattern. Using *stepwise regression,* the most relevant statistical metrics were retained and were used on the Statistical model. When compared to the other models, the Statistical Model has the best performance, particularly for the RMS metric.

Chapter 5

Conclusions

In this thesis, bitstream-based NR quality metrics for H-264/AVC encoded video, when transmitted over IP networks, were proposed and evaluated. Since video's perceived quality, on video communication systems, is mainly affected by encoding and transmission losses, the objective quality metrics focused on these two types of impairments. The models rely on information taken from the bitstream, namely quantized DCT coefficient data and information taken from the packet headers, and were of the no-reference (NR) type.

The results achieved for the objective quality assessment of encoded video (so, not considering packet losses) have shown that the linear model lead to the best performance, followed closely by the Sigmoid1 model.

As for the objective video quality assessment in IP networks, the results achieved have shown that the Statistical Model lead to the best performance. The model uses the information taken from the

packet headers to compute several statistical metrics that describe the packet loss pattern. It also takes into account the frame type of the packet losses, since this can also have a strong influence on the video perceived quality. Combining this model with the one derived in Chapter 3, results in a global metric that accounts for compression errors and transmission losses.

Although the Statistical Model has shown a good performance, there is still room for improvements. As previously mentioned, video decoders use error concealment techniques to prevent video degradation caused by packet losses. Some techniques work better than others. However, all of them can more efficiently conceal a loss when the video sequence doesn't have much temporal and/or spatial activity. Accordingly, it is expected that by better quantifying the video spatio-temporal activities, more accurate objective video quality metric could be developed at the expense of increased complexity.

Additionally, the video sequences used on the subjective quality tests are quite limited. For instance, intra-frame refreshing was not used during encoding so the impact of a packet loss may propagate till the last frame of the video sequence. Since video transmission on IP networks is prone to packet losses, the use of intra-frame refreshing is recommended. Also, the use of only intra-frame prediction, as error concealment technique in I-frames, may not be the best idea. If an I-frame has a high spatial activity the spatial intra-frame prediction may face some difficulties in performing the predictions. A database that allows the study of the impact of the different network and coding parameters would be extremely useful and could help to improve the Statistical model.

References

[Cisc08] Cisco Systems, “Preserving Video Quality in IPTV Networks”, *IEEE Trans. on Broadcasting*, vol. 55. No. 2, pp. 386–395, September 2008.

[Weng03] S. Wenger, “H.264/AVC over IP” *IEEE Trans. on CSVT*, vol. 13, no. 7, pp. 645-656, July 2003.

[WiMo08] S. Winkler and P. Mohandas, “The evolution of video quality measurement: from PSNR to hybrid metrics,” *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, September 2008.

[WSBL03] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, “Overview of the H.264 /AVC Video Coding Standard”, *IEEE Trans. on CSVT*, vol. 99, no 4, pp. 626-642, July 2003.

[BCQ11] T. Brandão, M. Chin, and M. P. Queluz, “From PSNR to perceived quality in H.264 encoded video sequences”, in proc. Of QoEMCS, Lisbon, Portugal, 2011.

[CBQ12] M. Chin, T. Brandão, and M. P. Queluz, “Bitstream-based quality metric for packetized transmission of H.264/AVC encoded video”, in *proc. Of IWSSIP, Vienna, Austria, 2012.*

[Roq09] L. Roque, “*Quality Evaluation of Coded Video”,* Tese de Mestrado, Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal, 2009.

[KGPL06] Y. Kukhmay, K. Glasman, A. Peregudov, A. Logunov, “Video Over IP Networks: Subjective Assessment of Packet Loss” *IEEE Tenth International Symposium on Consumer Electronics*, IEEE, 2006.

[KXMP06] S. Kumar, L. Xu, M. Mandal, S. Panchanathan, “Error Resiliency Schemes in H.264/AVC Standard” *Elsevier J. of Visual Communication & Image Representation (Special issue on Emerging H.264/AVC Video Coding Standard),* Vol. 17(2), April 2006.

[BrQu10] T. Brandão and M. P. Queluz, “No-reference quality assessment of H.264 encoded video,” *IEEE Trans. on CSVT*, vol. 20, pp. 1437, November 2010.

[BrQu08] T. Brandão and M. P. Queluz, “No-reference PSNR estimation algorithm for

H.264 encoded video sequences," in proc. of EUSIPCO, Lausanne, Switzerland, August 2008.

[YoZX09] F. You, W. Zhang, J. Xiao, "Packet Loss Pattern and Parametric Video Quality Model for IPTV", *Eighth IEEE/ACIS International Conference on Computer and Information Science*, IEEE, June 2009.

[YWXW10] F. Yang, S. Wan, Q. Xie and H.R. Wu, " No-Reference Quality Assessment for Networked Video via Primary Analysis of Bit Stream", *IEEE Trans. on CSVT*, Vol. 20, no. 11, November 2010.

[VQEG03] VQEG. "Final report from the video quality experts group on the validation of objective models of video quality assessment, phase II". Technical report, www.vqeg.org, August 2003.

[ITUT08] ITU-T, "Recommendation J.246 – Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference," 2008.

[ITU08] ITU-T, "Recommendation J.247 – Objective perceptual multimedia video quality measurement in the presence of a full reference", 2008.

[ITUT07] ITU-T, "Opinion model for video-telephony applications", ITU-T Recommendation G.1070 April 2007.

[JoAr10] J. Joskowicz, J. Ardao, "A parametric model for perceptual video quality estimation" Springer Science+Business Media, LLC 2010.

[BeMo10] B. Belmudez, S. Moller, "Extension of the G.1070 Video Quality Function for the MPEG2 Video Codec", *Second International Workshop on QoMEX*, IEEE, June 2010.

[EVS04] A. R. Reibman, V. A. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network", *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 327–334, Apr. 2004.

[BRK09] A. Bhat, I. Richardson, S. Kannangara, "A novel perceptual quality metric for video compression", Picture Coding Symposium (PCS) 2009, California, Illinois, USA, May 2009

[WP02] S. Wolf and M. Pinson, "Video Quality Measurement Techniques", NTIA Report 02-392, June 2002.

[PQR09] M. P. Queluz, T. Brandão and L. Roque, "Subjective Video Quality Assessement", http://amalia.img.lx.it.pt/~tgsb/H264_test/, 2008-2009.

[ITU98] ITU-R BT. 500-9, "Methodology for the subjective assessment of the quality of television pictures", 1998.

[BRQ09] T. Brandão, L. Roque, and M. P. Queluz. "Quality assessment of H.264/AVC

encoded video". *In proc of Conference on Telecommunications*, Santa Maria da Feira, Portugal, April 2009.

[SNTD09] F. De Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro and Y. Ebrahimi, "Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel" in *proc. of QoMEX*, S. Diego, USA, 2009.

[ITUT99] ITU-T, "Subjective video quality assessment methods for multimedia applications", *Recommendation ITU-R P 910,* September 1999

[MoRu03] D. C. Montgomery, G. C. Runger, *"Applied Statistics and Probability for Engineers"*, Third Edition, 2003.