



João Ricardo Simões Soares

Quality Evaluation of 3D Video Subject to Transmission Channel Impairments

Dissertação de Mestrado

17 de Setembro de 2013



UNIVERSIDADE DE COIMBRA



FACULDADE DE CIÊNCIAS E TECNOLOGIA DA
UNIVERSIDADE DE COIMBRA

MESTRADO INTEGRADO EM
ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

**Quality Evaluation of 3D Video Subject to
Transmission Channel Impairments**

João Ricardo Simões Soares

Júri

Presidente: Professor Doutor José Manuel Fernandes Craveirinha

Vogal: Professor Doutor Vítor Manuel Mendes da Silva

Orientador: Professor Doutor Luís Alberto da Silva Cruz

Coimbra, 17 de Setembro de 2013

Acknowledgments

In the following lines, I would like to declare my sincere acknowledgments to everyone that have helped me during this research work.

I would like to begin by expressing a special gratitude to my supervisor, Professor Dr. Luís Cruz, who have always believed in my researching and working capabilities and was always available to discuss all technical aspects and new ideas for the work, and have put a lot of effort on reviewing extensively this essay and the published papers, which lasted several days. He has launched me to the first international conference where, for the first time ever, I made an oral presentation in English to a large scientific audience. He also encouraged me to participate in the Plenoptics training school, in Sundsvall – Sweden on June 2013; I will never forget this new experience and the new fellowships I have made, neither the “lack of night” and sun light entering in my hostel room at 3 a.m. For all his support, I am forever grateful.

I would like to thank Professor Dr. Pedro Assunção, and his research team, the opportunity of working as researcher in the project *3DVQM – 3D Video Quality Monitor* (IT/LA/P01131/2011), whose subject is addressed in this thesis. I would like to thank Instituto de Telecomunicações for the support and laboratory facilities that gave me conditions to accomplish this work. I also would like to thank my lab colleagues.

For the participants of the subjective assessment sessions – João Marcos, João Campos, Geovana Espírito Santo, Pedro Rocha, Daniela Reis, Pedro Bento, Sofia Ferreira, Diana Guardado, Fernando Cruz, Nuno Almeida, Válder Liberado, João Santos, Luís Castela, João Nunes, Herman Dumby, Marco Manaia, Pedro Ramos, Tiago Ferreira, Wojciech Hajduczenia, Dr. Luís Cruz, Diogo Pinto, Sérgio Araújo, José Lopes, Jorge Proença, Diana Mourão, Diogo Sereno, José Nunes, Solange Silva, Rui Barbosa, Luís Raposo, Luís Antunes, Bruno Gil, António Simões, and João Gante – I hope you all have enjoyed watching 3D videos without special glasses. Thank you all for the patience in evaluating the quality of 40 videos; that meant a lot for my work.

Also, I want to express my gratitude to my graduation and non-graduation friends, for all good and bohemian moments outside the working environment.

Finally, a special thanks to my family to whom I owe all I have become: my parents, my brother and sister-in-law, and my beautiful nieces Maria Inês and Matilde.

Abstract

This thesis presents a research work on no-reference quality assessment models for use in future 3D video broadcast applications over packet-loss-prone channels, such as Internet Protocol networks. The objective is to study the state-of-the-art quality measures for 3D video, described in the scientific literature, and to propose new empirical quality evaluation methods specific for packet-loss effects on the 3D quality of experience (QoE).

Empirical models with different granularities for outputting scores are proposed: sequence-level, GOP-level, and frame-level. The methods' main input parameters are the average Packet-Loss-Rate, types of affected frames (I, P or B) and some pixel-domain descriptors of the spatiotemporal complexity and packet-loss concealment artifacts, of both the texture and depth packet-streams. The outputs are the Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarities Index Model (SSIM) – or their averages in the case of coarse granularities – of the DIBR-synthesized view of interest. The modeling approach used to obtain the functional relationship between inputs and outputs was based on neural networks, as they support a large number of inputs, with low computational complexity after training. In order to train these models with an acceptable generalization, hundreds of transmission simulations were performed with different packet-loss-rates and mean-burst-lengths. Most of the models achieve very high accuracy: Pearson Linear Correlation Coefficient (PLCC) over 0.95 between estimates and the data. In order to measure the correlation between the sequence-level objective quality scores and the corresponding differential mean opinion score (DMOS) values, a set of subjective tests was performed involving 35 participants. Results show that DMOS correlates very well with the estimated DMOS from the average PSNR of the synthesized view (PLCC of 0.98) and reasonably well with the estimated DMOS from the average SSIM (PLCC of 0.89).

The proposed methodologies and empirical models can be used in an industrial setting, deployed in real-time quality monitoring systems of service and network providers, in order to identify and classify the severity of the QoE degradations due to transmission losses (e.g. PT Inovação ArQoS[®]).

Keywords: 3D Video, texture-plus-depth, depth-image-based rendering, transmission losses, Quality of Service, Quality of Experience, no-reference quality assessment, low complexity, neural networks.

Resumo

Esta dissertação apresenta um trabalho de investigação no âmbito de modelos sem referência para avaliação de qualidade de vídeo 3D, no formato textura-mais-profundidade. No futuro, espera-se que sinais de vídeo 3D venham a ser difundidos neste formato, em redes de pacotes sujeitas a perdas, como por exemplo redes IP, substituindo progressivamente a difusão 2D. O objectivo é estudar o estado-da-arte no que respeita aos modelos de qualidade de vídeo 3D, publicados na literatura científica, e propor novos modelos empíricos específicos que permitam quantificar de forma perceptualmente consistente os efeitos das perdas de pacotes na qualidade de experiência (QoE).

Modelos empíricos com diferentes granularidades são propostos nesta dissertação: sequência-a-sequência, GOP-a-GOP e trama-a-trama. Os principais parâmetros de entrada destes modelos são a taxa média de perda de pacotes, os tipos de tramas afectadas (I, P ou B) e alguns descritores de complexidade espaço-temporal e de degradação por disfarce de erros, definidos no domínio do pixel, para a textura e a profundidade. As saídas dos modelos são o PSNR e o SSIM – ou a sua média ao longo de uma série de tramas consecutivas – da vista sintetizada em consideração. A abordagem seguida para a modelação baseia-se em redes neuronais, pois estas suportam um número elevado de entradas e o seu desempenho, após a fase de treino, é de baixa complexidade computacional. Para treinar estes modelos com boa capacidade de generalização, foi necessário efectuar milhares de simulações com diferentes taxas de perdas de pacotes e diferentes comprimentos médios de rajadas. A maioria dos modelos revelou uma correlação elevada (PLCC acima de 0.95) entre as saídas e os valores objectivo. Além disso, foi efectuado um conjunto de testes de avaliação subjectiva envolvendo 35 participantes, com o objectivo de medir a correlação entre os valores de qualidade objectiva médios das sequências e os valores DMOS. Os resultados revelam que os valores de opinião média diferencial (DMOS) exibem uma correlação elevada com a DMOS estimada a partir do PSNR médio da vista sintetizada (PLCC de 0.98) e uma correlação média-elevada com a DMOS estimada a partir do SSIM médio (PLCC de 0.89).

Assim, as metodologias e modelos empíricos propostos podem ser usados em ambiente industrial, implementados em sistemas de monitorização de tempo real de QoE dos operadores de serviço e de rede, para identificação e classificação da severidade de degradações devido a perdas nas transmissões (e.g. PT Inovação ArQoS[®]).

Palavras-chave: Vídeo 3D, textura-mais-profundidade, síntese baseada em mapas de profundidade (DIBR), perdas na transmissão, qualidade de serviço, qualidade de experiência, avaliação de qualidade sem referência, baixa complexidade, redes neuronais.

Contents

Chapter 1 - Introduction.....	1
1.1 Context and motivation	1
1.2 Objectives and main contributions	4
1.3 Outline of the thesis	5
Chapter 2 - DIBR-based 3D video.....	7
2.1 System overview and advantages of DIBR-based 3D video.....	7
2.2 Estimation and representation of depth maps.....	8
2.3 Transmission schemes in packet networks.....	10
2.4 DIBR view synthesis – mathematical description.....	12
2.5 Analysis and illustration of occlusions and hole-filling in DIBR	14
2.6 Stereo/multiview displaying and free viewpoint video	16
Chapter 3 - Objective quality assessment of 3D video	17
3.1 Overview and classification.....	17
3.2 Media-layer FR image quality models	20
3.3 Summary of the newest 3D video quality assessment metrics.....	21
Chapter 4 - H.264/AVC encoding and Gilbert-Elliot model for packet-loss simulations.....	23
4.1 Introduction	23
4.2 Frame types and GOP structure.....	23
4.3 Macroblocks and slices.....	24
4.4 JM Reference Software and RTP packets	24
4.5 Gilbert-Elliot model for packet losses	25
4.6 Transmitter simulator	26
4.7 Error concealment of JM decoder	28
Chapter 5 - Empirical packet-layer models for synthesized view quality assessment	29
5.1 Context, objective and procedures.....	29
5.2 Video dataset and encoder setting parameters.....	30

5.3	Sequence- and GOP-level quality assessment.....	30
5.4	Single-input models	31
5.5	Neural networks based models	33
5.6	Modeling the effect of texture losses.....	37
Chapter 6 - Subjective quality assessment of 3D video.....		43
6.1	Introduction	43
6.2	Test Environment and subjects.....	43
6.3	DSCQS session: procedures and results.....	44
6.4	2D-3D Pair-Comparison session: procedures and results	46
Chapter 7 - Empirical hybrid models for frame-level synthesized view quality assessment....		47
7.1	Context, objective and procedures.....	47
7.2	Input packet-layer binary parameters	48
7.3	Input media-layer histogram-based descriptors.....	49
7.4	3D-VQM Architecture 1: single neural network model	51
7.5	3D-VQM Architecture 2: double neural network model.....	52
7.6	Averaging scores with visual attention models	53
Chapter 8 - Conclusion		55
Annex A: H.264/AVC syntax overview		57
Annex B: Trace-file Generator (Matlab script)		59
Annex C: Camera calibration parameters.....		60
Annex D: Content description of 3D video used.....		61
Annex E: Packet-layer parameter extractor (C++)		63
Annex F: GOP structure detector (C++)		65
Annex G: Packet-Layer specifications and overall quality of the 3D videos used in the subjective tests		66
Annex H: Discontinuity measure at macroblock edges (Matlab script)		67
Annex I: Examples of the media-layer quality monitor performance for scheme <i>DOL</i>.....		68
Bibliography		69

List of Figures

Figure 1.1 – Global consumer IP traffic forecast by Cisco	2
Figure 1.2 – Example of a frame extracted from a 3D video in texture-plus-depth format	3
Figure 1.3 – Examples of low quality visual stimuli.....	4
Figure 2.1 – Example of a virtual view synthesis with the 1 st frame of Champagne Tower with (VSRS 3.5).	7
Figure 2.2 – General concept of a 3D video broadcasting system based on texture-plus-depth video.	8
Figure 2.3 – Stereo camera geometry (seen from top), and depth representation. [21].....	9
Figure 2.4 – Three consecutive depth frames from Kendo, to exemplify temporal inconsistencies from estimation. 10	
Figure 2.5 – MPEG-2/RTP/UDP/IPv4 encapsulation [22].	10
Figure 2.6 – Examples of approaches for texture-plus-depth coding and transmission.	11
Figure 2.7 – Luminance of the left and right synthesized views of the Ballet’s 1 st frame.	15
Figure 2.8 – Left and right synthesized views of the Ballet’s 1 st frame	15
Figure 2.9 – Mapping functions for left and right views of the Ballet’s 1 st frame – 450 th line.	15
Figure 2.10 – Left: Cinema spectators wearing passive-polarized glasses as they watch a preview of the movie Avatar (photo: REUTERS). Right: left-right view spatial separation with a lenticular system (from Wikipedia).	16
Figure 3.1 – Media-Layer models categorized accordingly to the availability (or lack) of the reference.....	17
Figure 3.2 – No-reference quality monitor which uses a FR-estimated empirical model.	18
Figure 3.3 – Reference (left) and distorted (right) 120 th frame of a synthesized view of Poznan CarPark.....	21
Figure 4.1 – Frame reference relationships within a group of pictures with 7 frames [22].....	24
Figure 4.2 – Two-state Markov process for the Gilbert-Elliot model	25
Figure 4.3 – Frame order of an open GOP with 15 frames [22].....	27
Figure 4.4 – Example of three GOPs of a decoded packet-loss-impaired video with four fixed-size slices	27
Figure 4.5 – Spatial concealment of four contiguous slice loss of an I-frame	28
Figure 5.1 – Experiment setup for depth-only loss approach.....	30
Figure 5.2 – Sequence-level <i>PSNR</i> and <i>SSIM</i> estimation with a single-input parameter (depth-only loss approach). 32	
Figure 5.3 – GOP-level <i>PSNR</i> and <i>SSIM</i> estimation with a single-input parameter (depth-only loss approach).....	33
Figure 5.4 – Partially detailed two-layer network corresponding to equation (5.5).	34
Figure 5.5 – Sequence- and GOP-level <i>PSNR</i> and <i>SSIM</i> estimation with three input parameters	35
Figure 5.6 – Sequence- and GOP-level <i>PSNR</i> and <i>SSIM</i> estimation with 10 input parameters	36
Figure 5.7 – GOP-level <i>PSNR</i> and <i>SSIM</i> estimation with 10 input parameters of Table 5.2.....	37
Figure 5.8 – Experiment setup for extended approaches with texture losses (schemes <i>SIM</i> , <i>SCA</i> and <i>SIM</i>).....	37
Figure 5.9 – Venn diagrams representing texture and depth packet-losses for four different schemes.	38
Figure 5.10 – Sequence- and GOP-level <i>PSNR</i> and <i>SSIM</i> estimation for scheme <i>SCA</i>	40
Figure 5.11 – Sequence- and GOP-level <i>PSNR</i> and <i>SSIM</i> estimation for scheme <i>FRA</i>	40
Figure 5.12 – Sequence- and GOP-level <i>PSNR</i> and <i>SSIM</i> estimation for scheme <i>SIM</i>	41
Figure 6.1 – Grading console for DSCQS (left) and voting console for PC (right), written in Portuguese.	44
Figure 6.2 – Presentation structures: (a) DSCQS session, (b) 2D-3D Pair-Comparison session.	44
Figure 6.3 – Ranking criteria for distorted 3D videos.....	45
Figure 6.4 – DMOS vs. <i>PSNR_{Seq}</i> (left) and DMOS vs. <i>SSIM_{Seq}</i> (right) for the 20 evaluated videos.	46
Figure 6.5 – Voting results for 2D-3D Pair Comparison Test.	46

Figure 7.1 – Example of initial-loss classification given the same error pattern as Figure 4.4.....	48
Figure 7.2 – Overall structure of the hybrid 3D Video Quality Monitor (3D-VQM Architecture 1).....	51
Figure 7.3 – Target slice SSIM versus predicted slice SSIM with the 3D-VQM Architecture 1, for all schemes.....	52
Figure 7.4 – Overall structure of the hybrid 3D Video Quality Monitor (3D-VQM Architecture 2).....	52
Figure 7.5 – Target slice SSIM versus predicted slice SSIM with the 3D-VQM Architecture 2.....	53
Figure 7.6 – Target slice SSIM versus predicted slice SSIM with the 3D-VQM Architecture 2.....	54
Figure A.1 – H.264/AVC syntax overview (Figure 4.19 from [56]).....	57
Figure A.2 – H.264/AVC syntax overview (Figure 5.1 from [56]).....	58
Figure A.3 – Performance of the media-layer 3D-VQM (Architecture 1) for scheme <i>DOL</i>	68
Figure A.4 – Performance of the media-layer 3D-VQM (Architecture 2) for scheme <i>DOL</i>	68

List of Tables

Table 3.1 – Classification of objective quality measurement methods in the context of IPTV [28].	17
Table 3.2 – PLCC performance comparison of some state-of-the-art image and 2D video quality algorithms [32].	19
Table 3.3 – Full-reference methods for 3D video quality assessment.	22
Table 3.4 – Reduced-reference methods for 3D video and image quality assessment.	22
Table 3.5 – No-reference methods for 3D video and image quality assessment.	22
Table 5.1 – Encoder setting parameters of the videos used in the depth-only loss approach.	31
Table 5.2 – Input parameters for 3D video quality assessment with neural network accurate models.	35
Table 5.3 – PLCC of the quality prediction accuracy of the models obtained for depth-only loss approach.	36
Table 5.4 – Encoder setting parameters of the videos used in schemes <i>SIM</i> , <i>FRA</i> and <i>SCA</i> .	38
Table 5.5 – Input parameters for 3D video quality assessment with neural network accurate models (all schemes).	39
Table 5.6 – PLCC of the quality prediction accuracy of the models obtained for all schemes.	41
Table 6.1 – Logistic fitting coefficients and PLCC of the plots of Figure 6.4.	45
Table 7.1 – Input parameters for slice-wise quality assessment of the synthesized view.	49
Table A.1 – Camera calibration parameters of the videos used in simulations.	60
Table A.2 – Nearest and farthest depth values of depth maps used in simulations.	60
Table A.3 – Content description of Balloons	61
Table A.4 – Content description of Kendo	61
Table A.5 – Content description of Champagne Tower.	61
Table A.6 – Content description of Newspaper	62
Table A.7 – Content description of Lovebird.	62
Table A.8 – Content description of Poznan CarPark	62
Table A.9 – Specifications and overall quality of the Balloons sample set	66
Table A.10 – Specifications and overall quality of the Champagne Tower sample set	66
Table A.11 – Specifications and overall quality of the Kendo sample set	66
Table A.12 – Specifications and overall quality of the Poznan CarPark sample set	66

List of Acronyms and Abbreviations

2D	Two-dimensional
3D	Three-dimensional
3DTV	Three-dimensional Television
3D-VQM	Three-dimensional Video Quality Monitor
AFR	Affected-Frame Rate
CABAC	Context-adaptive binary arithmetic coding
CSV	Conventional Stereo Video
DCT	Discrete Cosine Transform
DIBR	Depth-Image-Based Rendering
DMOS	Differential Mean Opinion Score
DSCQS	Double Stimulus Continuous Quality Scale
FPGA	Field-Programmable Gate Array
FR	Full-Reference
FRA	Frame-compatible texture-plus-depth scheme
FVV	Free viewpoint Video
GOP	Group of Pictures
H.264/AVC	Advanced Video Coding
IDR	Instantaneous Decoding Refresh
IETF	Internet Engineering Task Force
IPTV	Internet Protocol Television
IPv4	Internet Protocol version 4
ITU	International Telecommunication Union
JM	Joint Model H.264/AVC Reference Software
JSVM	Joint Scalable Video Model
LB	Number of Lost Bytes
MBL	Mean Burst Length
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
MTU	Maximum Transmission Unit
MVC	Multiview Video Coding
MVD	Multiview Video plus Depth
NAL	Network Adaptation Layer
NR	No-Reference

P2P	Peer-to-Peer
PLCC or R	Pearson Linear Correlation Coefficient
PLR	Packet Loss Rate
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantizer Parameter
RFC	Request for Comments
RGB	Red-Green-Blue
RR	Reduced-Reference
RTP	Real-time Protocol
SCA	Scalable texture-plus-depth encoding scheme
SIM	Simulcast texture-plus-depth scheme
SSIM	Structural Similarity Index Model
SVC	Scalable Video Coding
TB	Total Number of Bytes
TOF	Time of Flight
TS	Transport Stream
UDP	User Datagram Protocol
VOD	Video On Demand
VQEG	Video Quality Experts Group
VSRS	View Synthesis Reference Software

Chapter 1 - Introduction

«It would take an individual over 5 million years to watch the amount of video that will cross global IP networks each month in 2017».

Cisco: The Zettabyte Era — Trends and Analysis (May 29, 2013)

1.1 Context and motivation

As the demand for digital 3D video is increasing, in part due to the growing offer of 3D cinema immersive feature films, it is expected that in a few years broadcasts of 3D Television (3DTV) will become part of our everyday life, progressively replacing 2D television broadcasts. Some small steps have already been made towards this goal: since 2008, some live sport events have been broadcasted in 2D-frame-compatible side-by-side stereo format. Most current 3D video solutions are based on the rendering and displaying of multiplexed left and right views. Special passive anaglyphic, polarized, or active-shutter glasses channel each view to the corresponding human eye, inducing the stereo parallax that allows depth perception. However, some experts believe that the breakthrough for 3D television will only come when glasses won't be needed to that purpose, with the use of autostereoscopic or even the holographic displays.

Moreover, the fast growth of the 2D – and, in the future, 3D – television broadcasted over the Internet (IPTV) and Video-on-Demand (VOD) services will drastically increase the amount of data traffic exchanged in the supporting networks. According to a recent forecast published by Cisco [1], annual global IP traffic has increased more than fourfold in the past five years, and is expected to increase threefold over the next five years, surpassing the zettabyte (10^{21} bytes) threshold by the end of 2015. As shown in Figure 1.1, consumer Internet video traffic (P2P traffic excluded) will represent 73% of all consumer Internet traffic in 2017, up from 60% in 2012. If we include the video exchanged through P2P file sharing systems, this ratio will surpass 80%. This huge traffic growth will raise the bar for the coding and compression efficiency requirements and force the upgrade of network infrastructures and transmission protocols to support higher bandwidths; otherwise the network quality of service (QoS) will decrease as a result of increased traffic, leading to congestion in routers and, consequently, packet losses, specially in the cases of real-time protocols (like RTP) that do not allow retransmissions. Severe jitter, defined as the temporal variations of the propagation delay of consecutive packets, also leads to events similar to packet losses as a result of the uselessness of the video data received outside of its usability time-window.



Figure 1.1 – Global consumer IP traffic forecast by Cisco

The goal of any multimedia delivery system is to ensure the best video end quality possible. As described in [2], in an IPTV or a VOD service, a single digital video item might pass from content provider to service provider to network provider before reaching the end viewer. The content provider wants to ensure that their video, which he created with a given quality level, is not further degraded when delivered to the final consumer; service providers want to guarantee that the video they got from the content provider has sufficient quality and that the network provider does not degrade it significantly, in order to protect their brand images.

Deploying video quality monitors at the set-top box receiver, or even at some node in the transport network, allows managing the streaming services by adjusting dynamically some of the transmission (and coding) parameters in order to deliver content at adequate perceived quality level, while optimizing resource usage [3], [4]. For instance, for large bit-error wireless channels, video can be transcoded at the edge of the wired network by decreasing the coding bitrate: in other words, by lowering the video quality with more aggressive source coding. This strategy would leave more bits available for improved error protection schemes, with stronger Forward Error Correction codes. In an automatic quality controller, unequal protection schemes may be used to improve protection for the most important packets in detriment to the less important ones, effectively assigning different priorities to different data [5], [6]. As an example of the application of this principle, in the case of 3D video in the texture-plus-depth format (exemplified in Figure 1.2 and explained in chapter 2), packets carrying texture data may be labeled as more important than those transporting depth information [7]. In reliable transmission scenarios, the continuous measurement of the jitter allows the receiver to adjust the buffer size and buffering times, and to request the retransmission of lost packets, at least the most important



Figure 1.2 – Example of a frame extracted from a 3D video in texture-plus-depth format, which is the scope of this thesis. It consists of the texture frame (left) and the per-pixel depth information in the form of a depth map (right).

ones. Finally, if all these quality control mechanisms proved to be unsuccessful, providers may adopt variable billing schemes, according to the quality of the received contents. It is thus clear that digital video quality monitoring is becoming a more and more important for operation and management of (3D) video delivery systems. Currently, PT Inovação uses the quality monitoring system ArQoS[®] for voice calls/connections over different kinds of network technologies (fixed/mobile/IP) [8], and in the future is expected to add new modules related to (3D) video quality.

Researchers working on objective visual quality assessment aim to create methodologies and models capable of predicting the perceived quality of a visual stimulus, by humans [9]. However, perceived quality is subjective, and so varies from observer to observer. Thus, in order to obtain a mean opinion score (MOS) of the perceived quality of a given visual stimulus (e.g. image, video), a sizeable quantity of human observers (at least 15) is shown that stimulus and are asked to evaluate it on an opinion grading scale [10], [11]. Scores outputted from an objective model used for assessing the perceived quality of that stimulus (among other similar assessed stimuli) must correlate with MOS. Yet, measuring perceived quality of video which has been subject to rare and unpredictable events like packet losses is a challenging quality assessment problem [2].

To distinguish a good quality video from one with bad quality, it is common to aggregate technical factors such as the spatiotemporal resolution and the presence of distortions or artifacts due to compression and transmission impairments like: blurring, ringing, jagged motion, ghosting, freezing, blocking or slicing effects (Figure 1.3). In 3D video based on the texture-plus-depth format, we also need to take into account the visual comfort and distortions induced by rendering of virtual views using depth-image-based rendering (DIBR). Even if the overall quality of the 2D texture is excellent, distortions affecting the depth information, inconsistent rendering of virtual views, sub-optimal display conditions, and the vergence–accommodation conflict may lead to degradation in the perception of depth, visual fatigue and severe mental confusion, in other words, a terrible quality of experience (QoE) for the viewer.

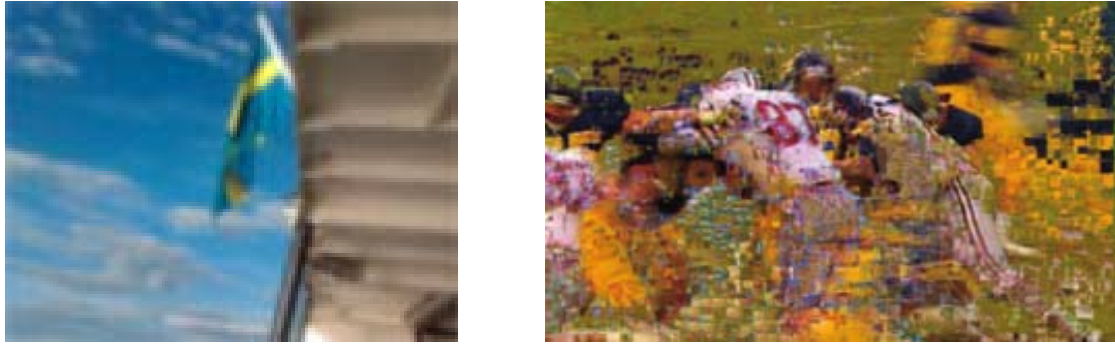


Figure 1.3 – Examples of low quality visual stimuli: (left) frame from a blurred and magnified low-resolution video; (right) frame affected by severe loss in decoded video.

Although one can consider the 3D video as a simple extension of the 2D video, the classical methods for 2D video quality assessment are not well suited to fully assess 3D video quality. The MOS obtained from subjective assessment sessions becomes multidimensional: in addition to the image quality itself, depth perception and visual comfort of the displayer's technology would play a major role in the subjects opinion [12], so that global subjective quality scores result from a combination of those factors weighted according to the subject's criteria. As a result of these specificities, the problem of estimating 3D video overall quality is a lot more complex than the corresponding problem of evaluating 2D video quality.

1.2 Objectives and main contributions

This thesis presents a study on objective 3D video quality measures able to quantify the effects of packet losses on the quality of the 3D video (in the texture-plus-depth format) decoded after impaired transmission, with emphasis on losses of the depth information. To simplify the problem and decouple coding and transmission effects, the original uncompressed quality as well as compression/coding related quality degradations of both texture and depth information are not taken into account in this work. Thus, only the effects of the artifacts affecting synthesized virtual views which are due to packet losses are studied and modeled.

The objectives of this thesis, as well as its main scientific contributions, are summarized in the following four topics:

- Conduct a bibliographic search on recent methodologies and models published in recent scientific journals and conference proceedings dealing with topics related to the evaluation of quality of texture-plus-depth 3D video. Prepare a comparative analysis of the most promising works from the point-of-view of their applicability to the subject matter of this thesis.

- Propose and study new approaches for objective quality assessment of DIBR-synthesized 3D video subject to packet losses, averaged on a temporal window, based on no-reference packet-level and low-complexity empirical models.
- Conduct a subjective assessment study, focused on depth-only impaired 3D videos, in order to measure the correlation between the objective quality scores from the previous models and the MOS values, and evaluate their efficiency.
- Propose and study new approaches for frame-level objective quality assessment of DIBR-synthesized 3D video subject to packet losses, based on no-reference media-layer low-complexity descriptors and packet-layer parameters.

During the development of the work and writing of this thesis, two articles describing no-reference models for 3D video quality prediction [13], [14] were written, published and presented in two scientific conferences: 9th Conference on Telecommunications and IEEE ICC 2013. Other two short papers describing the methodologies and results of packet-layer models for DIBR-synthesized 3D video quality assessment (chapters 5 and 6 of this thesis) and an article that describes some hybrid models for frame-level objective quality assessment (chapter 7) are currently being prepared for submission.

1.3 Outline of the thesis

In chapter 2, the state-of-the-art of the DIBR-based 3D video is presented, with emphasis on depth map estimation issues, coding and transmission solutions and schemes, and the mathematical description of the virtual view synthesis used in DIBR.

Chapter 3 provides information on the state-of-the-art methods for multimedia quality assessment, using a methodology based on the classification of the methods according to the use (or lack) of original un-encoded video information.

In chapter 4, the fundamentals of the H.264/AVC video coding and error concealment technique used in the Joint Model (JM) Reference Software are explained. The Gilbert-Elliot model to obtain packet-loss traces used in the transmitter-simulator software is also explained.

Chapter 5 describes the packet-layer model proposed to evaluate synthesized 3D video subject to packet losses and presents results of its application to several test cases, at sequence- and GOP-level.

In chapter 6, the procedures followed in and the results of the subjective assessment study of the impaired 3D videos are presented.

Chapter 7 describes the procedures and results of the frame-level hybrid quality assessment model for impaired synthesized 3D video.

Finally, chapter 8 concludes this thesis by summarizing the results obtained and suggesting future research activities to be performed on the same subject.

Chapter 2 - DIBR-based 3D video

2.1 System overview and advantages of DIBR-based 3D video

There are three main types of 3D scene representations:

- **Volumetric** representation: is the natural extension from 2D to 3D, where instead of pixels the data are represented by voxels. This representation is mainly used in computer-generated graphics for gaming or medical purposes, and is out of the scope of this thesis.
- **Multiview** representation: the 3D visual information is represented with a minimum of two views, known as conventional stereo video (CSV) and used for example in 3D cinema, to dozens of views. There are a few possible coding and transmission approaches suitable for use with this type of representation: (a) simulcast of one or more 2D views; (b) frame-compatible formats by multiplexing two or more views into single composite 2D frames, such as left-right, top-down, interlaced or anaglyphic; (c) multi-view coding (MVC), probably the most efficient approach for this representation, by exploiting redundancies between views – inter-view prediction – and encode them into a single bitstream. As the details of these approaches are not in the scope of this work, please refer to [15] for further information.
- **Depth-Based** representation: is a simple extension from 2D video (texture), enhanced with its associated per-pixel depth information, known as depth maps, used to synthesize virtual views in real-time (exemplified in Figure 2.1). Similarly to the case of multiview video, we can use different coding and transmission arrangements, from the simple video-plus-depth simulcast to the extended multiview-plus-depth format (MVD).

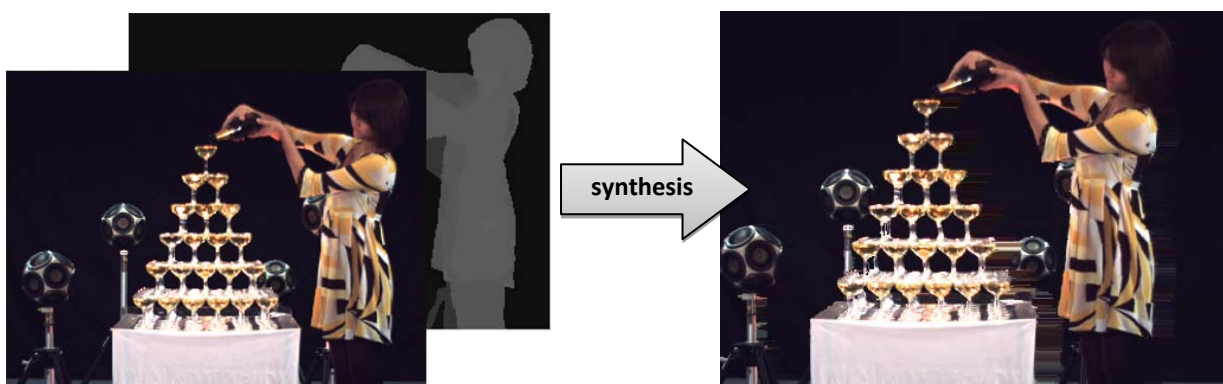


Figure 2.1 – Example of a virtual view synthesis with the 1st frame of Champagne Tower. Left: texture-plus-depth frame from the original view. Right: corresponding frame of the virtual view of a camera placed to the right of the original camera, synthesized with the MPEG View Synthesis Reference Software (VSRS 3.5).

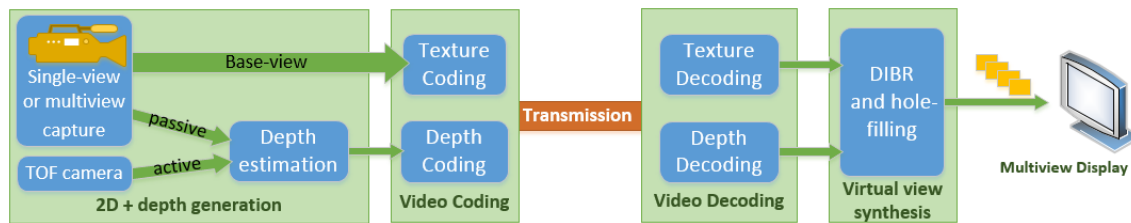


Figure 2.2 – General concept of a 3D video broadcasting system based on texture-plus-depth video.

The depth-based representation is believed to be the choice for future 3D video storage and broadcast systems, because it comprises a number of advantages [16], such as:

- Possibility of a customized 3D experience, with either stereoscopic or autostereoscopic displays, according to personal preferences of the viewer (e.g. more or less presence).
- Simulation of head-motion parallax, creating a look-around effect and eliminating “shear-distortions”.
- Efficient compression: noise-free depth maps can be compressed up to 25% of the texture bitrate [17], which makes 3D video format based on texture complemented with depth information compatible with the current transmission bandwidth.
- Absence of photometrical asymmetries, in terms of brightness, contrast or color between the views, as they are all synthesized from the same original 2D texture.

The overall 3D Video delivery chain is presented in Figure 2.2, which shows the main stages: 2D-plus-depth content generation, coding, transmission, decoding, DIBR-synthesis and display.

2.2 Estimation and representation of depth maps

Depth estimation techniques can be classified into two categories:

- **Optical active** estimation [18], [19], using time-of-flight (TOF) cameras such as ZCam, Kinect and Fotonics. Typically infrared light is projected from the TOF camera, is reflected by the objects in the scene, being captured by the receiver sensor to rebuild the depth information based on the flight time. Issues regarding the low spatial resolution, low depth range, low reflectance of dark objects, and inaccurate depth estimation on object boundaries make this approach seldom used for the scope of 3DTV.
- **Image-based passive** estimation [20]–[22], which has been widely studied. The estimation is based on a frame-wise pixel or block matching between different views, or in case of direct 2D-to-3D conversion, the estimation is based on motion information from two consecutive frames. As the computational complexity is very high, hardware-based solutions are the most promising for real-time depth estimation.

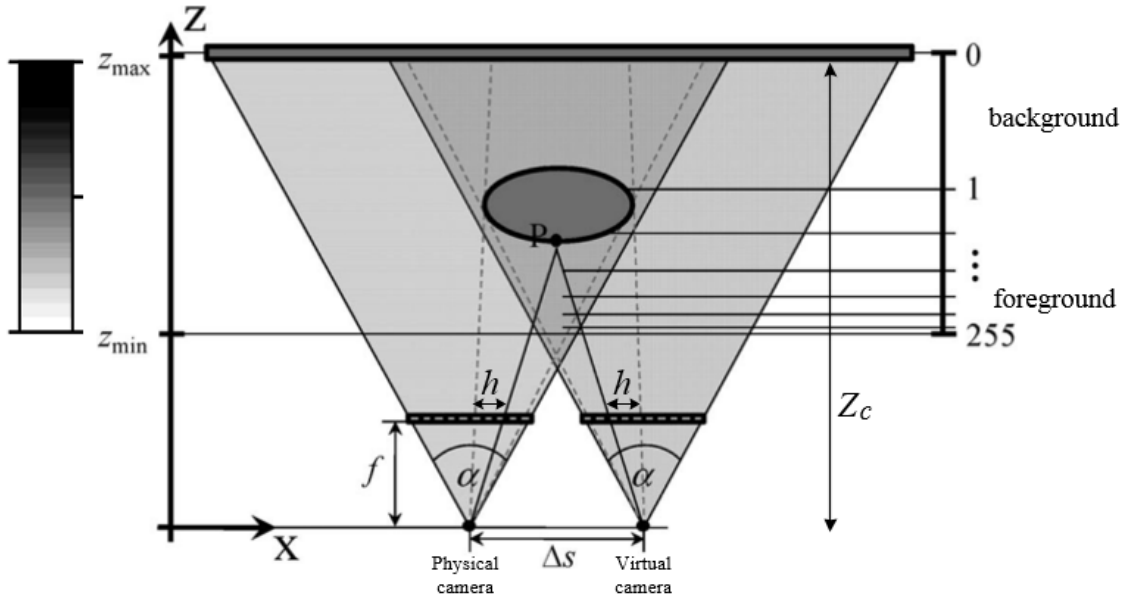


Figure 2.3 – Stereo camera geometry (seen from top), and depth representation. [23]

Depth maps are typically encoded as greyscale image sequences, with the same spatiotemporal resolution of the texture. Each pixel represents the inverse of the depth value of the co-located texture pixel, quantized to 8-bits. The inverse relation between the depth and the map value is given by equation (2.1):

$$I_d(z) = \text{round} \left[255 \cdot \left(\frac{1}{z} - \frac{1}{z_{\max}} \right) / \left(\frac{1}{z_{\min}} - \frac{1}{z_{\max}} \right) \right] \quad (2.1)$$

where z_{\min} and z_{\max} are defined as the minimum and maximum distances recorded in a video sequence, and z is the real depth of the pixel, in some user-defined unity.

As we can see in Figure 2.3, this representation offers a finer distance resolution for closer objects, and coarser resolution for farther objects. This differentiation of resolution with respect to distance is in accord with the human perception of depth.

As the state-of-the-art passive depth estimation algorithms operate on a frame-by-frame basis, depth maps often exhibit temporal inconsistencies or noise, as shown in Figure 2.4. These inconsistencies reduce the effectiveness of the temporal prediction used in state-of-the-art video encoders like H.264/AVC, resulting in harder-to-code depth maps. Nevertheless, provided a high enough bitrate is used during the coding, such temporal inconsistencies generally do not degrade the quality of the virtual views if the texture pattern, located in the depth noise regions, is uniform.



Figure 2.4 – Three consecutive depth frames from Kendo, to exemplify temporal inconsistencies from estimation.

2.3 Transmission schemes in packet networks

Broadcasted video encoded using H.264/AVC is packetized into Network Abstraction Layer (NAL) packets, which contains information of a frame or a portion of a frame (slice), and then encapsulated into 188 byte MPEG-2 Transport Stream (TS) packets. To transport the TS packets over IP networks, they are grouped into seven packets and encapsulated into RTP/UDP/IPv4 datagrams [24], as shown in Figure 2.5.

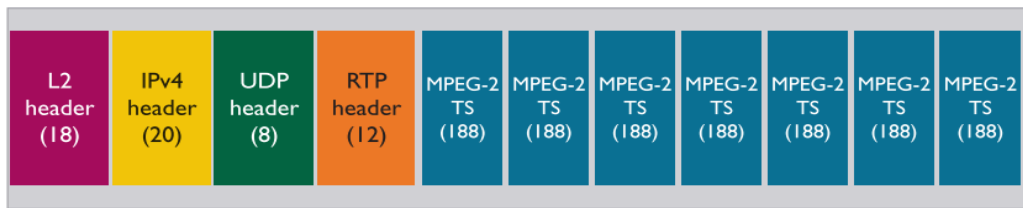


Figure 2.5 – MPEG-2 TS/RTP/UDP/IPv4 encapsulation [24].

Different coding and transmission schemes may be adopted for texture-plus-depth video, with different purposes and levels of backward compatibilities. For instance, Figure 2.6 (a) shows the simplest mode of simulcasting texture and depth in independent bitstreams. A 3D set-top box decodes the texture and depth (two different program streams) simultaneously, then synthesizes the virtual views, and sends them to a stereo/multiview display; a 2D set-top box only decodes one channel (the texture is the natural choice) and sends the output to a 2D display. Thus, the simulcast approach is fully backward compatible, but has the poorest efficiency, mainly because the encoder does not exploit redundancies between texture and depth, like the similarities of texture and depth motion vectors. Figure 2.6 (b) is a scheme similar to the frame-compatible CSV where in this case a composite frame is made up of the texture side-by-side with the depth map. This arrangement is signaled with Supplementary Enhancement Information (SEI) messages multiplexed in the bitstream, and may not be fully backward compatible if the 2D set-top box does not interpret correctly the SEI messages in order to decode and present in the

display only the texture (possibly after an interpolation for restoring the original frame resolution). Moreover, this scheme may be more efficient if the motion vectors of both texture and depth are shared. However, the main drawback is that the H.264/AVC encoder will encode both texture and depth in the same NAL units, or in the same bitstream, leading to poor transmission loss resilience. Figure 2.6 (c) shows a more efficient scheme with scalability, in which the texture is encoded as the base layer and the depth is encoded as an enhancement layer, with shared motion vectors from the texture. The bitstreams are separated, making this approach backward-compatible with 2D set-top boxes, which discard the unknown depth information. This last approach is more error-resilient because the enhancement-layer may be corrupted and the base-layer still may be correctly decoded.

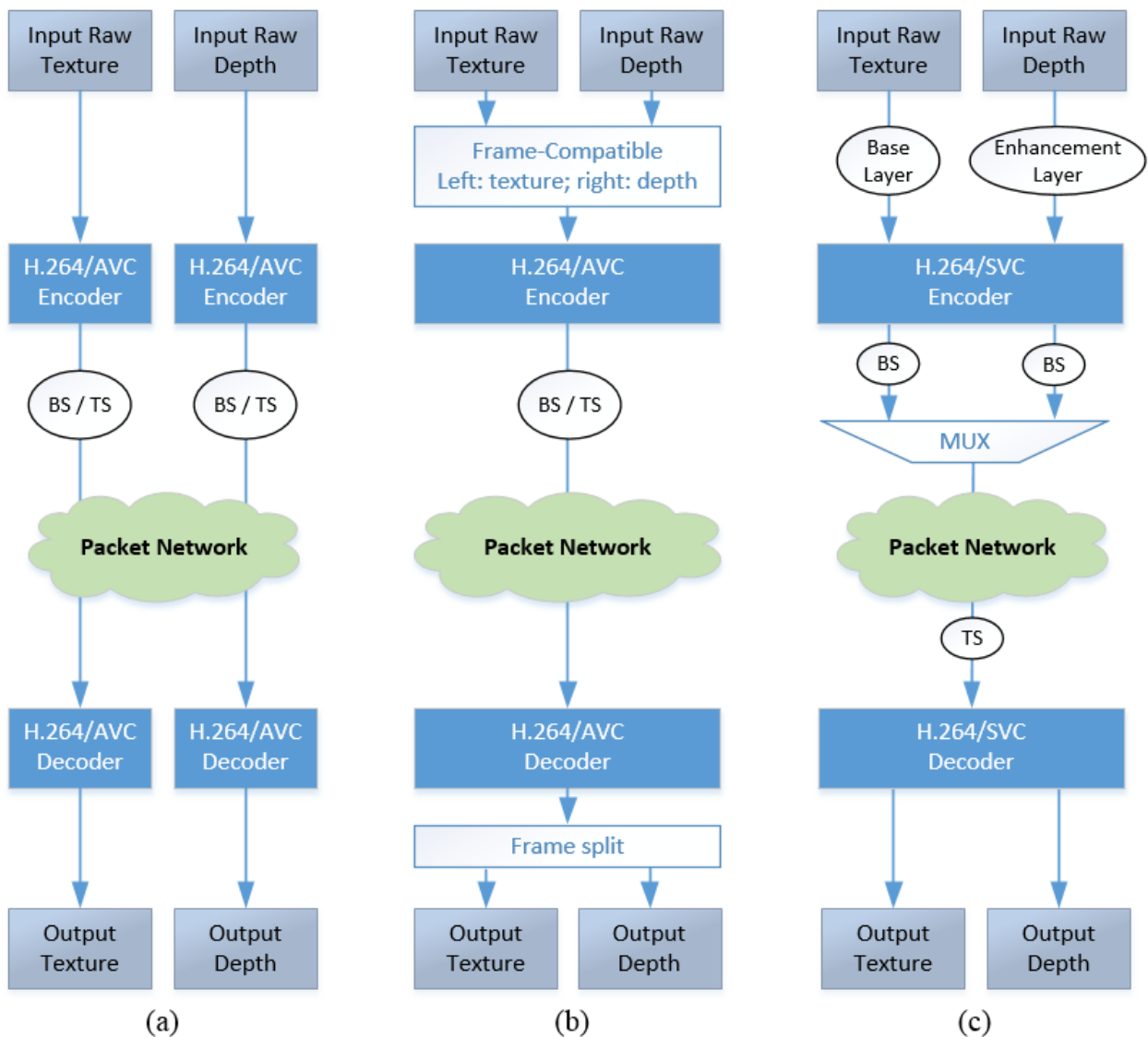


Figure 2.6 – Examples of approaches for texture-plus-depth coding and transmission: H.264/AVC Simulcast (a), H.264/AVC Frame-Compatible (b), and H.264/SVC with Scalability (c).

2.4 DIBR view synthesis – mathematical description

In this section, the mathematical 3D image warping formalism [16] describing the view synthesis procedure will be briefly explained. Consider a system of two coplanar cameras separated by the distance Δs (baseline), and an arbitrary 3D space point $P = (x, y, z)^T$ with the projections $p_l = (u_l, v_l)^T$ and $p_r = (u_r, v_r)^T$ in the original (left) – resp. virtual (right) view, as shown in Figure 2.3. Let the 3x3 matrices R_l and R_r , and the 3x1 vectors t_l and t_r define the rotation and translation that transform the space point from the world coordinate system into the camera coordinate system of the left and right cameras (extrinsic parameters); let the two upper triangular 3x3 matrices A_l and A_r specify the intrinsic parameters of the cameras, according to:

$$A = \begin{bmatrix} \text{focal length } x & \text{radial distortion} & \text{principal point } x \\ 0.0 & \text{focal length } y & \text{principal point } y \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (2.2)$$

Let z_l and z_r describe the scene depth in each camera coordinate system, and let's assume, without losing generality, that the world coordinate system equals the camera coordinate system of the left camera, such as $R_l = I_3$ and $t_l = [0 \ 0 \ 0]^T$. Then, the two perspective projection equations that map point P to points p_l and p_r of the views are:

$$z_l \tilde{p}_l = A_l [R_l | t_l] \tilde{P} \Leftrightarrow z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = A_l [I | 0] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Leftrightarrow z_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = A_l \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.3)$$

$$z_r \tilde{p}_r = A_r [R_r | t_r] \tilde{P} \Leftrightarrow z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = A_r [R_r | t_r] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Leftrightarrow z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = A_r R_r \begin{bmatrix} x \\ y \\ z \end{bmatrix} + A_r t_r \quad (2.4)$$

where the tilde variables symbolize point coordinates in homogeneous notation. Inverting the final form of equation (2.3), leads to:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = z_l A_l^{-1} \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} \Leftrightarrow P = z_l A_l^{-1} \tilde{p}_l \quad (2.5)$$

Substituting equation (2.5) into equation (2.4) leads to the classical affine disparity equation, which defines the depth-dependent relation between corresponding points in two images of the same scene:

$$z_r \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = z_l A_r R_r A_l^{-1} \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} + A_r t_r \Leftrightarrow z_r \tilde{p}_r = z_l A_r R_r A_l^{-1} \tilde{p}_l + A_r t_r \quad (2.6)$$

Taking into account simplifications on the geometry of the problem, namely that the intrinsic parameters of both cameras are equal (except the horizontal shift h of the respective principal points) and that the movement of the right virtual camera is restricted to be translational in the x -axis with respect to the original left camera, it follows that $A_r = A_l = A$, $R_r = R_l = I$ and $z_l = z_r = Z$.

Therefore, Equation (2.6) reduces to:

$$\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} + \frac{A t}{Z} + \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix}, \quad \text{with} \quad t = \begin{bmatrix} \eta \cdot \Delta s \\ 0 \\ 0 \end{bmatrix} \quad (2.7)$$

Finally, the pixel coordinates u_r and v_r of the virtual view can be obtained as:

$$\begin{cases} u_r = u_l + \eta \frac{\alpha_u \cdot \Delta s}{Z} + h \\ v_r = v_l \end{cases}, \quad \text{with} \quad h = -\eta \frac{\alpha_u \cdot \Delta s}{Z_c} \quad (2.8)$$

in which Z_c is the convergence distance and α_u represent focal length in multiples of the pixel width, defined as $\alpha_u = f \cdot m_u$, where m_u is the scale factor relating pixels to distance. $\eta = 1$ if the virtual view is to the right of the original camera (as in Figure 2.3), and $\eta = -1$ otherwise.

Equation (2.7) gives the warping pixel mapping from the left view (original) to the right view (virtual). Note that the depth values Z can be retrieved from depth maps by inverting equation (2.1), according to:

$$Z = \left[\frac{I_d(z)}{255} \left(\frac{1}{z_{\min}} - \frac{1}{z_{\max}} \right) + \frac{1}{z_{\max}} \right]^{-1} \quad (2.9)$$

2.5 Analysis and illustration of occlusions and hole-filling in DIBR

Taking into account the assumptions and simplifications that lead to equation (2.8), one can analyze the DIBR as a one-dimensional pixel mapping, performed over the frame line-by-line. Let $C(u)$ be a generic pixel array representing a single horizontal line extracted from a texture frame; let $V(u^*)$ be the pixel array representing the co-located single line from the synthesized virtual view. The pixel mapping can be interpreted as follows:

$$V(u^*) = C(u) \Leftrightarrow V(u + d(u)) = C(u) \quad (2.10)$$

where $d(u)$ represent the disparity in units of pixels, computed from the depth information, as previously discussed. We can also define a generic function as follows:

$$u^* = f(u) = u + d(u) \quad (2.11)$$

This function is, by rule, non-injective, meaning that different values for u (i.e. different pixels) may be mapped to the same value u^* . This originates occlusions, in which foreground objects are interposed in the line of sight of background objects, leading to *pixel superposition by replacement*. Furthermore, this function is also, by rule, non-surjective, meaning that the pixel mapping may not cover the entire codomain of u^* . This originates deocclusions, in which foreground objects move away from background previously occluded objects, leading to “holes”, as shown in Figure 2.7. However, the view-synthesis algorithm must guarantee that the virtual view does not show “black holes” from deocclusions (Figure 2.8), and must put foreground objects in front of background objects during occlusions. In other words, it must assure an injective and surjective mapping: that’s why this algorithm is typically performed iteratively, in a specific rastering direction that depends on the position of the virtual view camera with respect to the original view camera. Figure 2.9 shows an example of pixel mapping for a left and right virtual view, according to the generic function (2.11).

The quality of the hole-filling technique plays a major role in the quality of the virtual synthesized views, and a lot of effort has been put into this topic to find good solutions [25]–[27]. Typical solutions are based on the replication of the background pixel next to the foreground. Spatiotemporal low-pass filtering on these deoccluded regions is helpful to reduce noise and smooth the “unknown” texture. Besides, it has to be assured that coded depth maps object edges retain their original quality, i.e. they don’t show compression artifacts. That may be very difficult to accomplish with the state-of-the-art encoders such as H.264/AVC, calling for solutions to improve the depth map coding efficiency such as that of [28].



Figure 2.7 – Luminance¹ of the left and right synthesized views of the Ballet's 1st frame, without hole-filling (the green color serves here only to highlight no-filled deoccluded regions, or holes).



Figure 2.8 – Left and right synthesized views of the Ballet's 1st frame, with the hole-filling technique developed for the work in [29].

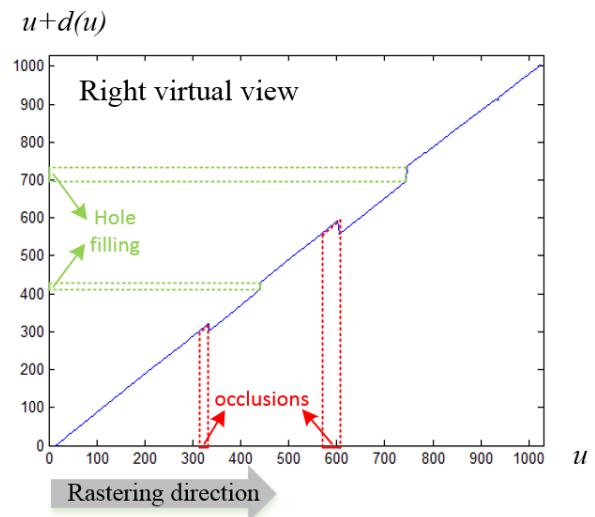
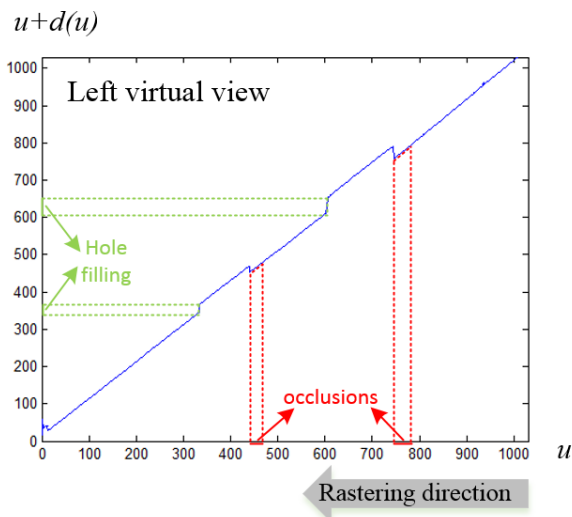


Figure 2.9 – Mapping functions for left and right views of the Ballet's 1st frame – 450th line.

¹ Luminance (Y) is the monochromatic representation of an RGB color image. Color is added by two chrominance components: U and V. These components are defined as linear combinations of the RGB color channels.

2.6 Stereo/multiview displaying and free viewpoint video

Once the desired views are synthesized, they are ready to be presented to the viewer in a 3D display. There are two main technologies for 3D displaying: (a) stereoscopic, currently used in 3D cinema and widely available on the market for the common consumer; (b) autostereoscopic/multiview, which is not yet well popularized and is much more expensive than the first type. The main difference between the two types is that the stereoscopic display requires the user to wear special passive-anaglyphic (in the case of color-domain view multiplexing), passive-polarized (in the case of polarization-domain view multiplexing), or active-shutter glasses (in the case of temporal-domain view multiplexing), which channel each view to the corresponding human eye, allowing depth perception. On the other hand, autostereoscopic/multiview displays do not require glasses, because the different views are multiplexed in the spatial-domain by a parallax barrier or a lenticular system, and as long as the viewer is placed in the sweet-spot location each eye receives only one view. This article [30] provides a detailed state of the art in stereoscopic and autostereoscopic display technologies. Figure 2.10 shows cinema spectators wearing 3D passive-polarized glasses (left) and the concept of left-right view spatial separation with a lenticular system (right). Note that the spatial-domain multiplexing concept is not necessarily restricted to two views: for instance, in subjective tests described in chapter 6, a 9-view autostereoscopic display is used for presenting 3D content.

The texture-plus-depth 3D video (or its MVD extension) and DIBR synthesis can also be used in the context of free viewpoint video (FVV). In this case, a single virtual view of the viewer's preference is synthesized and then displayed in a 2D display².

3D video and FVV is expected to become part of our everyday life as soon as the QoE becomes good enough. Quality assessment aspects will be introduced in the next chapter.

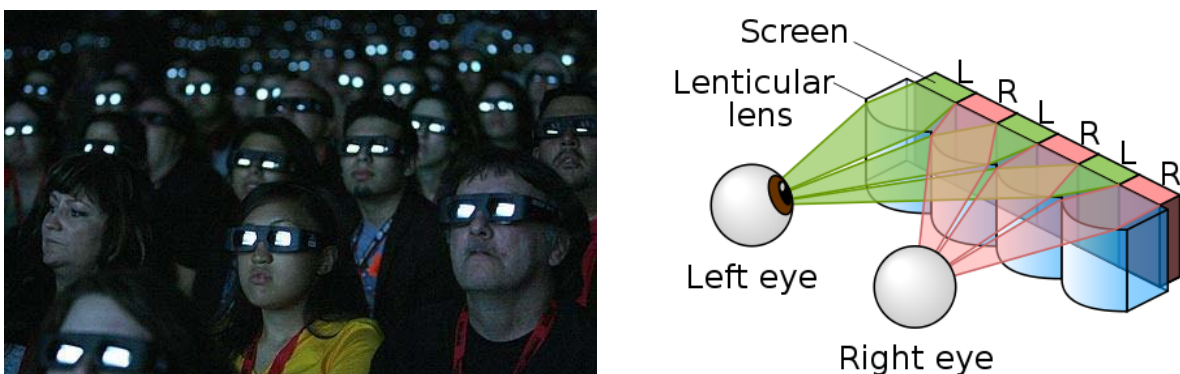


Figure 2.10 – Left: Cinema spectators wearing passive-polarized glasses as they watch a preview of the movie Avatar (photo: REUTERS). Right: left-right view spatial separation with a lenticular system (from Wikipedia).

² Check an example of 3D FVV in football stadium, available online on <http://www.youtube.com/watch?v=dvZa46SwjKc>

Chapter 3 - Objective quality assessment of 3D video

3.1 Overview and classification

Objective quality measurement methods for multimedia transmitted over packet-switch networks (e.g. Internet) have been classified into the following five main models (or layers) according to the input information used for quality assessment and the primary application [31]. These are parametric planning models, packet-layer models, bitstream-layer models, media-layer models, and hybrid models, as summarized in Table 3.1.

Media-layer models can be further categorized according to the availability (or lack) of the original non-distorted media – also known as the reference – to be compared with the impaired media [32]: full-reference (FR), reduced-reference (RR) and no-reference (NR), as shown in Figure 3.1. Full- and reduced-reference methods play an important role on evaluation of video systems in non-real-time scenarios, such as measuring the quality of multimedia encoders and transmission condition, at the development phase. While the presence of a reference image or information regarding the reference simplifies the task of quality assessment, practical applications of such algorithms are very limited in real-world scenarios, where the reference media is generally not available at the location/point where the quality computation is undertaken, or it is impossible to ensure its correct transmission to those locations by means of an ancillary reliable channel. Thus, NR methods are the best suited for these practical scenarios.

	Planning	Packet-Layer	Bitstream-Layer	Media-layer	Hybrid
Input information	Quality design parameters	Packet headers and codec information	Packet-layer and payload information	Pixel-domain	Combination of any
Primary Application	Network planning, terminal/application designing	In-service nonintrusive monitoring (e.g. network probe)	In-service nonintrusive monitoring (e.g. terminal-embedded operation)	Quality benchmarking	In-service nonintrusive monitoring

Table 3.1 – Classification of objective quality measurement methods in the context of IPTV [31].

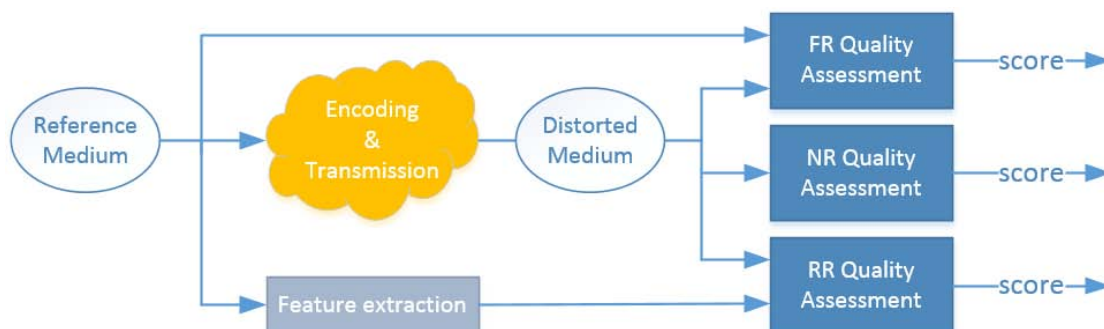


Figure 3.1 – Media-Layer models categorized accordingly to the availability (or lack) of the reference.

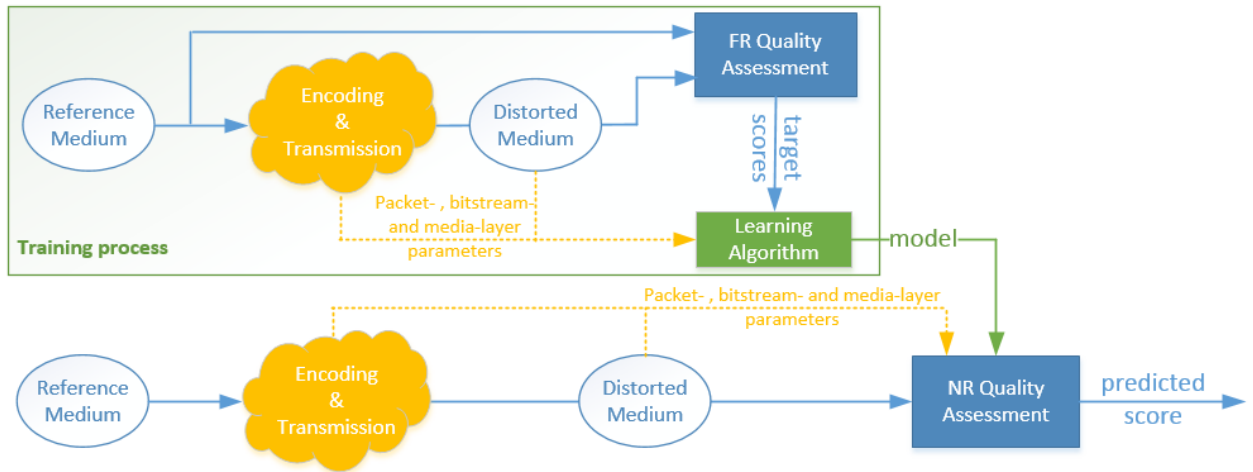


Figure 3.2 – No-reference quality monitor which uses a FR-estimated empirical model.

Moreover, some packet-layer, bitstream-layer and even no-reference media-layer methods use some input parameters, according to their layer of operation, to estimate full-reference media quality scores, such as the Peak Signal-to-Noise Ratio (PSNR) or the Structural Similarity Index (SSIM) [33]. This approach assumes a specific application, or even a specific type of impairment, and it is based on learning algorithms for fitting empirical parametric models (e.g. curve fitting, neural networks and support vector machines) that represent a functional relationship between input parameters and the estimated quality value. Once the model is correctly trained and validated, it can be deployed in a no-reference quality monitor, as shown in Figure 3.2. This is, in fact, a widely used approach in the work reported in the literature, and it was adopted in the work for this thesis as well (see chapters 5 and 7). The motivation for this approach is justified by the absence of other effective no-reference quality assessment methods that are accurate for a desired specific application (e.g. packet-loss events during transmission), and the need for low-complexity real-time quality estimation algorithms.

The performance and usefulness of a perceptual objective quality model depends on its correlation with subjective results. The most used metric for evaluating the performance of an objective video quality model is the Pearson Linear Correlation Coefficient (PLCC or R) between the subjective MOS values x_i and the MOS values y_i predicted from the objective model. For N data pairs (x_i, y_i) , with \bar{x} and \bar{y} being the means of the respective data sets, the PLCC (or R) is given by:

$$PLCC = R = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}} \in [-1, 1] \quad (3.1)$$

The ITU-R BT.500-11 [10] recommends the mapping between objective metric quality OQM scores and predicted MOS values y_i by means of a logistic function, defined as:

$$y = \frac{a_1}{1 + e^{a_2(OQM + a_3)}} \quad , \quad a_1, a_2, a_3 \text{ are fitting coefficients} \quad (3.2)$$

Due to the availability of several image and 2D video public databases, with common or different features (e.g. spatial resolution, types of distortion, severity of distortions, etc.), it is very difficult to perform realistic comparisons between different objective quality metrics [34]–[36]. Often, as a specific objective metric is designed to correlate well with subjective scores from a particular database, it is further found not to perform so well on data from other databases, as shown in Table 3.2. Furthermore, most quality metrics, like NR, are designed for certain types of image artifacts (e.g. blur, blocking ...) and are not well suited to evaluate multimedia content subject to other types of degradations. This issue is aggravated exponentially in the case of 3D video: recall that, as said in the introduction of this thesis, MOS of subjective assessment of 3D video becomes multidimensional, even if subjects find very difficult to distinguish “depth quality” from “visual comfort” [37]. Moreover, due to the lack of 3D image and video databases, most metrics are optimized for a very restricted set of videos – most of them are publicly available for a very short period of time.

The article [38] reports and compares some state-of-the-art quality metrics for 3D image and video. However, the authors recognize that it is very difficult (or even impossible) to compare the performance of two different 3D quality evaluation algorithms, even in a common dataset, due to practical reasons such as: intellectual property rights, different source 3D video formats (e.g. texture-plus-depth vs. left-right) and the unavailability of ground truth noise-free depth maps. The feature to be evaluated (depth quality, spatial quality ...) may also be different among different metrics. Therefore, the overall conclusion to close this section is: *we can compare only the comparable; every quality metric has its pros, cons, objectives and application scope.*

Database	VQEG	IRCCyN	EPFL-PoliMI	LIVE
PSNR	0.7683	0.4160	0.7351	0.5621
SSIM [33]	0.8215	0.5012	0.6781	0.5444
VQM [39]	0.8170	0.4850	0.8434	0.7236
MOVIE [40]	0.8210	0.4850	0.9210	0.8116
Yu <i>et al.</i> [41]	0.8170	0.7680	0.9470	0.8450
3D-SSIM [35]	0.8403	0.8194	0.9621	0.8353

Table 3.2 – PLCC performance comparison of some state-of-the-art image and 2D video quality algorithms [35].

3.2 Media-layer FR image quality models

The two most widely known and used media-layer models to evaluate the quality of a gray-scale image with reference to the original are the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [33]. PSNR can be defined as the logarithm of the inverse of the energy of the pixel-wise difference between the original and the distorted image (error signal):

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad , \quad \text{with} \quad MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (3.3)$$

where N is the number of pixels of the original and the distorted images, with pixel values x_i and y_i resp., and n is the number of bits per pixel (typically 8). The higher the PSNR, the better the quality. This formula can be used with video data by computing it in a frame-level basis usually applied to the luminance component, wherein the overall sequence score may result from either a simple or a weighted average of the frame PSNR values. For application-generic purposes, this measure has poor correlation with perceived image quality, since the MSE does not reflect the way that human visual systems perceive image degradation [42]. However, for specific impairments due to packet-loss events, some researchers claim that the PSNR can still be a good predictor of subjective video quality (in terms of MOS) [43], [44]. In chapter 5 this assumption is confirmed to uphold.

The human visual system is highly adapted to extract structural information from visual scenes. Therefore, a measurement of structural similarity (or dissimilarity) should provide a good approximation to perceptual image quality. Nonstructural distortions are distortions that do not modify the structure of objects in the visual scene. The SSIM evaluates image similarity based on three factors computed from the two images being compared: luminance $l(x,y)$, contrast $c(x,y)$, and structure $s(x,y)$, defined as:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad , \quad c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad , \quad s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (3.4)$$

where x and y are the reference and the distorted image luminance pixel values; μ , σ and σ_{xy} represent their mean, standard deviation and covariance; and C_1 , C_2 and C_3 are small constants added for numerical stability. These factors are combined to yield an overall similarity measure:

$$S \text{ SM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad , \quad \{\alpha, \beta, \gamma\} > 0 \quad (3.5)$$

$$\begin{cases} \alpha = \beta = \gamma = 1 \\ C_3 = C_2/2 \end{cases} \Rightarrow SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.6)$$

The score provided by the SSIM ranges between 0 and 1; the closer to 1, the higher the similarity of the distorted image to the reference image and so in this context the better the quality of the (distorted) image. Usually, this method is applied locally, within a local Gaussian circular-symmetric small window that moves pixel-by-pixel over the entire image, and then the local scores are averaged to produce the overall score. For color images, SSIM and PSNR scores can be computed for the luminance and chrominance components, in separate, and the final score can be weight-averaged. However, in this thesis, all SSIM and PSNR scores are computed only for the luminance component. Figure 3.3 shows an example of a video frame decoded from a bitstream affected by packet losses, together with its luminance's PSNR and SSIM scores.

3.3 Summary of the newest 3D video quality assessment metrics

This section reports on *some*³ state-of-the-art methods for 3D video quality assessment. Table 3.3, Table 3.4 and Table 3.5 exemplify resp. some FR, RR, and NR methods, explaining the application or the type of distortion to which the method is applicable, the features computed for measuring them, and the PLCC that informs how much the metric is correlated with subjective quality scores. According to the discussion in section 3.1, we cannot assume *a priori* that a method which has a higher PLCC is better than a method which has a lower PLCC. Even if the application and artifacts are similar, the video set used in subjective tests may be different, so they are not fully comparable. Values with a tilde indicate averages of the PLCC values, in the case that several PLCC values exist for various video sets.



Figure 3.3 – Reference (left) and distorted (right) 120th frame of a synthesized view of Poznan CarPark, extracted from a simulation with 20% PLR corrupted depth map. Red ellipses are only to point the most distorted zones. PSNR = 30.54 dB , SSIM = 0.938 (luminance component).

³ The word *some* is stressed here because due to the large number of methods reported in the scientific literature, it is impossible to cover them all; still an effort was made to gather methods that cover different applications.

Quality metric (Authors)	Application & Artifacts	Features used to measure the artifacts	PLCC
Solh <i>et al.</i> [45]	Depth map and colored video compression, depth estimation (passive stereo matching), and depth from 2D to 3D conversion.	Temporal outliers, temporal inconsistencies and spatial outliers, using ideal depth map estimation.	0.8942
Joveluro <i>et al.</i> [46]	Texture-plus-depth with scalable encoding (JSVM) at different QP. Quality of DIBR-synthesized views, using 2D metrics.	Distortion in the brightness and contrast distortion using an approximation (variances) weighted by the mean of each pixel block, of the luminance component of synthesized views.	~ 0.988
Sun <i>et al.</i> [47]	Stereoscopic 3D video encoded at different compression rates, or video-plus-depth rendered into left and right views.	Distortion for 8x8 block content (luminance and contrast) and distortion for block boundary of the luminance component of synthesized views.	~ 0.953
Yasakethu <i>et al.</i> [48]	Texture-plus-depth with scalable encoding (JSVM) at 1Mbps, packetized into 1400-byte packets and simulated the transmission over a packet network with different PLR.	After segmentation of depth planes: distortion of the relative distance within each depth plane, distortion in the consistency of each depth plane, and structural error of the depth.	0.8369

Table 3.3 – Full-reference methods for 3D video quality assessment

Quality metric (Authors)	Application & Artifacts	Features used to measure the artifacts	PLCC
Maalouf <i>et al.</i> [49]	JPEG symmetric and asymmetric coding of stereoscopic images.	Contrast sensitivity (spatial frequency and orientation) and coherence of cyclopean images (combination of locally matched stereo regions in a single global image).	0.981
Hewage and Martini [50]	Texture-plus-depth H.264/AVC encoded at different QPs and simulated the transmission over a packet network with different PLR.	Luminance, structure and contrast of the texture; edge-based structural correlation of the depth maps.	0.9273 (T) 0.9795 (D) (vs. FR)
Nur and Akar [51]	Texture-plus-depth with scalable encoding at different bitrates, with 80% / 20% bitrate allocation for texture and depth resp.	VQM [39] between the bilateral-filtered original depth map and the bilateral-filtered compressed depth map.	~ 0.913

Table 3.4 – Reduced-reference methods for 3D video and image quality assessment

Quality metric (Authors)	Application & Artifacts	Features used to measure the artifacts	PLCC
Sazzad <i>et al.</i> [52]	JPEG symmetric and asymmetric coding of stereoscopic images.	Blockiness and zero-crossing of edge, flat and texture areas, and average zero-crossing of plane and non-plane areas of the disparity.	0.960
Solh <i>et al.</i> [53]	Depth map and colored video compression, depth estimation (passive stereo matching), and depth from 2D to 3D conversion	Temporal outliers, temporal inconsistencies and spatial outliers, using a no-reference ideal depth map estimation.	0.916
Bosc <i>et al.</i> [54]	Structural distortion indicator for DIBR synthesized views due to imperfect matching at depth discontinuities, and hole-filling. DIBR algorithm benchmarking.	Contours detection and displacement estimation from central (texture) and synthesized view. Inconsistent displacement and motion vectors; new contours.	—
Mittal <i>et al.</i> [55]	Assess the comfort associated with viewing stereoscopic image and video.	Histogram-based features from disparity, disparity gradient maps, indicators of spatial activity, plus motion compensated disparity differences for videos.	~ 0.77 (SROOC) ⁴
Feitor <i>et al.</i> [13], [14]	Packet-Layer quality assessment of stereoscopic video subject to packet losses, in a frame-level basis.	Frame loss detection, frame type, and size estimate of the lost frame (in bytes).	~ 0.765 (vs. FR)

Table 3.5 – No-reference methods for 3D video and image quality assessment

⁴ This method does not provide PLCC, instead it provides Spearman Rank Order Correlation Coefficients (SROOC), which measure the prediction monotonicity of a metric, i.e. the degree to which the predictions of a metric agree with the relative magnitudes of the subjective quality ratings [79].

Chapter 4 - H.264/AVC encoding and Gilbert-Elliot model for packet-loss simulations

4.1 Introduction

In this chapter, some general concepts of H.264/AVC video encoding are explained. This encoder was chosen as it is currently the most popular and best performing encoder, with very high encoding performance. The details of its operation provided here cover just the basic information required to understand chapters 5 and 7 of this thesis. For more (technical) details we suggest the study of Recommendation ITU-T H.264 [56] as well as reference [57] which provides a easier to follow explanation of the technical recommendation. Finally, the IETF-RFC 6184 [58] defines the RTP Payload format commonly used together with this video codec. This chapter also explains the Gilbert-Elliot model for packet-loss simulations.

4.2 Frame types and GOP structure

Video is a sequence of images (frames) displayed at a certain rate (frame-rate), giving the perception of continuous movement for the viewer eyes. Typical frame-rates are 25 frames-per-second (fps) and 30fps. Frames can be encoded with a DCT-domain JPEG encoder [59], which explore the spatial redundancy. Contiguous frames from *natural* videos (as opposed to white-noise videos or sequences of uncorrelated images) are also temporally correlated. Thus, a good video codec can be able to efficiently explore the temporal redundancies between frames, by signaling reference frames and computing displacement (motion) vectors in a block-wise approach. Frames coded without reference to other frames are called I-frames; these are typically the least compressed frames. Rendered frames using information from at least one temporally preceding reference frame are called P-frames; reference frames are typically I- or P-frames. Finally, rendered frames using information from past and future reference frames are B-frames; these are typically the most compressed frames.

Frames are arranged into groups of pictures (GOP). A GOP includes the I-frame and all subsequent frames leading up to the next I-frame. Figure 4.1 shows the frame relationships inside a GOP. I-frames provide the reference point for decoding a received MPEG stream, and are very important in error propagation recovery. The length and structure of a GOP plays a very important role in the engineering task of balancing compression efficiency and error recovery. Compression can be controlled by fixing a quantization parameter (QP) or fixing a bitrate.

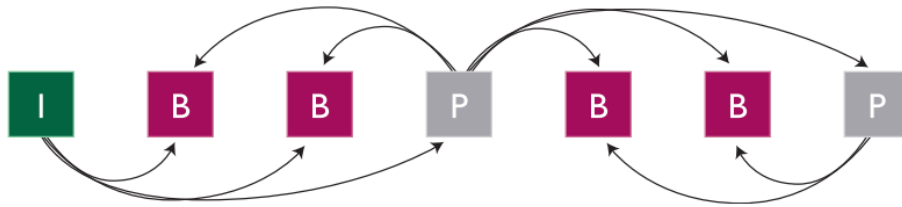


Figure 4.1 – Frame reference relationships within a group of pictures with 7 frames [24].

The GOP structure can be classified as *open* or *closed*: in the *open* mode, the I-frame can be a reference for the last B-frames of the previous GOP; in the *closed* mode, the inter-frame references are strictly confined inside the GOP, the last frame of the GOP is forced to be a P-frame and the I-frame is actually an *Instantaneous Decoder Refresh* (IDR) frame. In this thesis, only *closed* GOPs were used to prevent inter-GOP error propagation.

4.3 Macroblocks and slices

Macroblocks are groups of 16x16 pixels, acting as the elementary image partition unit. Each of these units carries information of the macroblock type (I, P or B), prediction modes or motion vectors, the Coded Block Pattern, the QP and residual data.

In its simplest arrangement, slices are formed by groups of consecutive macroblocks. Slices can be formed with fixed number of macroblocks or fixed number of bytes, and they play an important role in error resiliency as they confine the error propagation to a small area of the frame. Increasing the number of slices up to a reasonable amount (e.g. 10 per frame) increases the error resiliency on the one hand, which improves the overall quality of the video when transmitted in loss-prone channels; on the other hand, it reduces compression efficiency, since the inter-frame prediction is mostly confined to the slice area, and more overhead information is needed. Each slice is then packetized into a Network Abstraction Layer (NAL) unit.

Annex A provides an overview of the H.264/AVC syntax [57].

4.4 JM Reference Software and RTP packets

The Joint Model (JM) Reference Software, from Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VQEG is used in this thesis as the H.264/AVC codec. It is very popular among the scientific multimedia community, and it was designed mainly for research purposes. It is indeed a rather complex software, with hundreds of configuration parameters and combinations.

In this work, several video-plus-depth sequences were encoded separately. Each NAL unit, which contains all the information of just one slice, is packetized into a variable-length RTP packet. MPEG-2 TS packets, as shown in Figure 2.5, are not used here explicitly. In order to simulate realistic transmission schemes over IP-networks, the average stream packet size was set below the maximum-transmission-unit (MTU) size of 1500 bytes. Packets that exceed the MTU size are assumed to be split into IP-datagrams; losing at least one of them results in the loss of the entire slice.

4.5 Gilbert-Elliot model for packet losses

The impact of packet loss on real-time video streaming services can be studied from recorded measurement traces of traffic and loss patterns. To generate error patterns with similar characteristics as observed in measurements, for offline-simulations, stochastic models such as discrete-time Markov chain models can be used [60].

In this work, we use the Gilbert-Elliot model [61], which is a stochastic packet loss model based on a two-state Markov process (Figure 4.2). It is characterized by a good state ($X=0$), a bad state ($X=1$), and transition probabilities, p and q , between the two states, as response to events. We can define two events: (a) successful arrival of a packet, making the system transit to or remain in the good state; (b) packet loss detection or packet corruption, making the system transit to or remain in the bad state.

Gilbert-Elliot model memorizes only the previous state, thus the probability that the next expected packet will be lost, $P(X_{i+1} = 1)$, depends only on the current state of the system, X_i . This model is able to capture the dependence between consecutive losses, making it suitable for network transmission scenarios. Recall that bit errors or packet errors usually do not occur in a Bernoulli random fashion, but in bursts. Thus, the conditional transition probabilities can be calculated from two variables that characterize the transmission network: the *average packet loss rate (PLR)* and the *mean burst length (MBL)*; using the following equations derived from [62]:

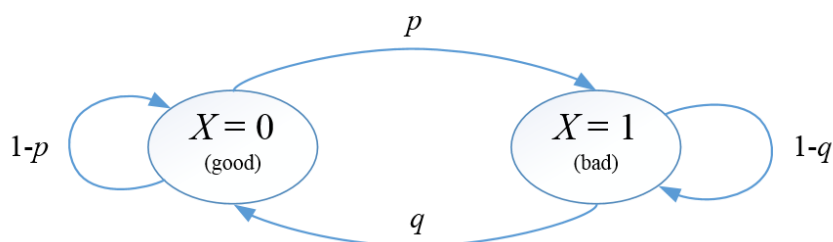


Figure 4.2 – Two-state Markov process for the Gilbert-Elliot model

$$p = P(X_{i+1} = 1 | X_i = 0) = \left[MBL \cdot \left(\frac{1}{PLR} - 1 \right) \right]^{-1} \quad (4.1)$$

$$q = P(X_{i+1} = 0 | X_i = 1) = \frac{1}{MBL} \quad (4.2)$$

provided that:

$$\frac{1}{PLR} > 1 + \frac{1}{MBL} \quad , \quad 0 < PLR < 1 \quad , \quad MBL \geq 1 \quad (4.3)$$

This model was implemented in Matlab, with the script provided in Annex B. The output is an error trace file, with 10000 characters, where the character ‘1’ means a lost packet and ‘0’ a successfully received one. Different combinations of (PLR , MBL) were used, to make various trace file patterns, with PLR ranging from 0.1% to 20% in nonlinear steps, and MBL from 3 to 7 in linear steps. The chosen range of values of PLR meets the typical range of values used by researchers: 0.1% loss is (most of the time) imperceptible for the viewer, and 20% loss usually leads to severe degradation of the video quality, so there is no added value in increasing beyond this rate. The chosen values for MBL provides some typical wired-network scenarios for short burst loss patterns – wireless networks probably require higher values – and meets the limitations of the JM decoder, whose robustness for handling very long burst losses is weak⁵.

4.6 Transmitter simulator

The transmitter-simulator software [63] used in the work described in the next chapters, corrupts the bitstream by discarding some RTP packets, according to a given error pattern file created as previously discussed. A modification was done to the source code, in order to output some packet-layer relevant parameters about the bitstream corruption process, namely:

- RTP packet size, in bytes, of all received and lost packets;
- Frame number, from slice header;
- Slice type (I, P or B), from slice headers;
- Received/lost (0/1) inferred from RTP header sequence-number discontinuities.

⁵ Due to the limitations and programming bugs of the error concealment of JM decoder, the maximum possible loss burst length was forced to the number of slices used minus one. Most coded videos use fixed 8 slices per frame, so the maximum burst length was forced to 7. In this way, a single frame is never lost entirely. Still, the simulations are believed to be realistic enough.

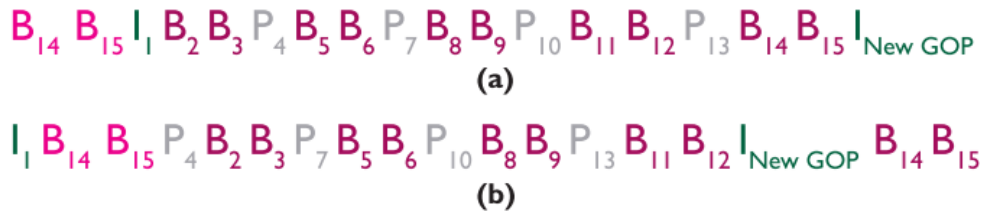


Figure 4.3 – Frame order of an open GOP with 15 frames [24]. (a) The encoder input order and decoder display order are the same, but (b) the transmission order and is different.

Note that the transmitter simulator does not shuffle the packets: they are ordered according to the transmission order, which is different from the displaying order after decoding, as exemplified in Figure 4.3. In real transmission scenarios, the packet reception order may be altered, but the de-jitter buffering and the sequence-number of the RTP packets can be used to put them back into the correct order. Figure 4.4 shows an example of a decoded packet-loss-impaired video with a closed-GOP structure and four fixed-size slices. Green slices represent no-affected slices; red slices represent self-lost and concealed slices; yellow slices represent error-propagated impaired slices; red-yellow striped slices represent the combination of the previous two situations. Note the impairment extension in the first two GOPs due to losses in the reference I-frames and P-frames. Losses in B-frames do not propagate, as shown in the third GOP.

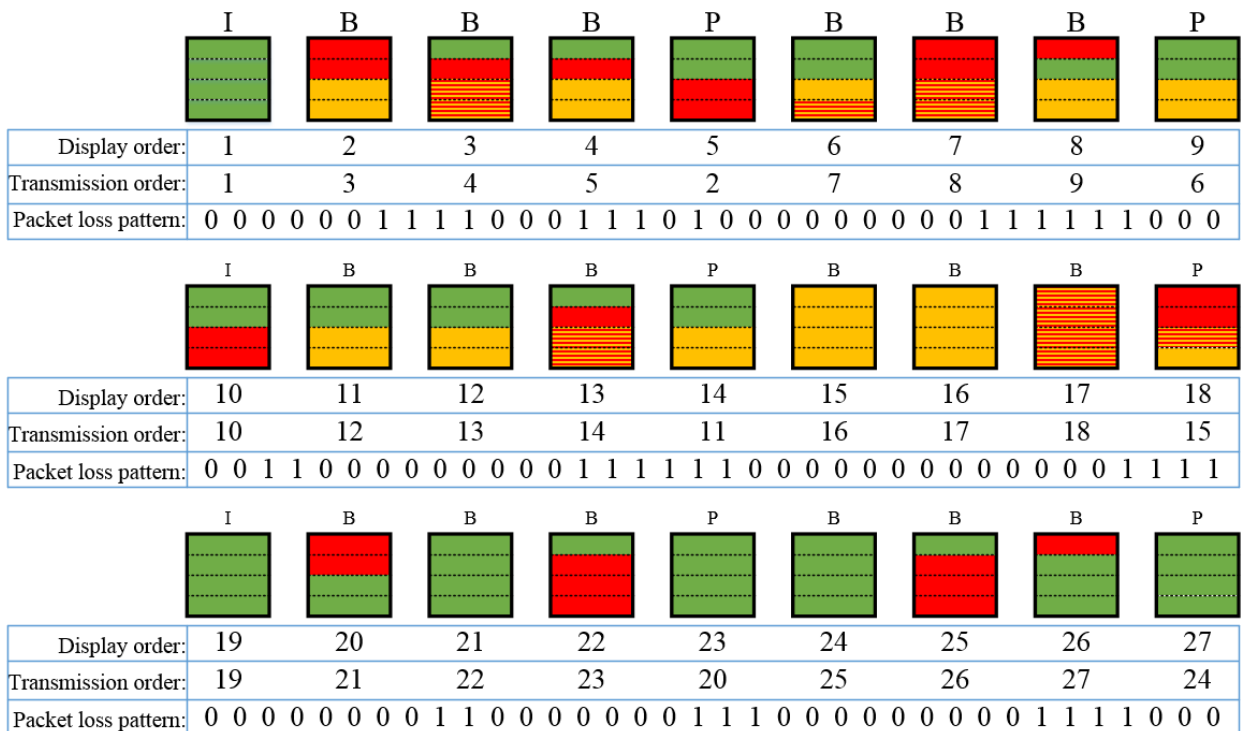


Figure 4.4 – Example of three GOPs of a decoded packet-loss-impaired video with four fixed-size slices.

4.7 Error concealment of JM decoder

When packets are lost or corrupted during transmission, the decoder tries to conceal the effects of these losses on the decoded video by the application of some recovery technique. The most common and simplest one is copying the slice from the previous correctly received and decoded frame; this technique is called frame-copy. Several sophisticated error concealment algorithms have been proposed in the literature [64]–[68]. Underlying the design of this type of concealment methods is a typical engineering search for a trade-off between the concealment quality and the computational cost of the operations required. As the error concealment methods are non-normative, a video decoder designer can adopt its own error concealment algorithm. The effect of packet-losses on the final video quality is thus decoder-dependent. JM decoder version 15.0, used in the work of this thesis, adopts intra-spatial concealment for I- and IDR-frames, and temporal concealment with motion compensation for P- and B-frames [69], [70].

In I- and IDR-frames, pixels of lost macroblocks are interpolated – or extrapolated – from the boundaries of the correctly received and decoded adjacent ones. Note that the quality of this technique is very poor for large slices or when a burst of slices is lost, as shown in Figure 4.5. If the entire I-frame is lost, then a copy of the last decoded frame from the previous GOP is used.

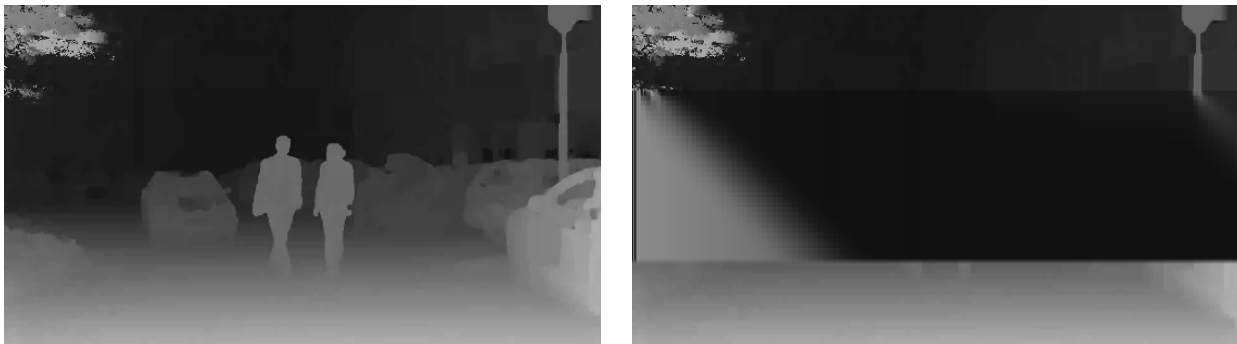


Figure 4.5 – Spatial concealment of four contiguous slice loss of an I-frame (Poznan CarPark depth frame no. 181). Left: correctly received and decoded version; right: impaired decoded version.

In P- and B-frame concealment, the decoder begins by evaluating the overall temporal activity of the correctly received slices. If the activity is low, slices are concealed by a direct copy of the co-located slices from the closest reference frame. Otherwise, two steps are performed: (a) estimation of the motion vector of the missing macroblocks from available motion information of its spatial or temporal neighbors; (b) use of the estimated motion vectors to find out the corresponding macroblocks in the nearest reference frame, and use them to substitute the missing macroblocks. This technique performs well when the number of slices is high; however block boundary artifacts become more and more visible as the error propagates throughout the GOP.

Chapter 5 - Empirical packet-layer models for synthesized view quality assessment

5.1 Context, objective and procedures

This chapter describes an empirical method for the modeling and evaluation of the quality of 3D video, in the texture-plus-depth representation, based on the use of some statistical network performance indicators. The objective of the method proposed is to allow the estimation of an objective quality score (e.g. PSNR, SSIM...) of a synthesized view, averaged over a temporal window of a GOP or a few continuous GOPs, from a set of parameters obtained from the packets' headers. Once this empirical model is obtained, it can be used in in-service non-intrusive monitoring systems, such as PTInovação ArQoS[®], with the model implemented as part of a network probe or based on data collected by a network probe. The output of the 3D video quality model should be useful for service and network providers, as it potentiates fast detection of failures and vulnerabilities of the core network as well as localized congestions that affect the end-user 3D video QoE. Providers and network operators can then adopt actions to correct undesirable situations, either at the broadcast center by changing video encoding settings or by adjusting the consignment of data transport resources; additionally, knowledge about the transmission impairments magnitude and location can be used to improve the operation and management of the transport network.

At a first approach, only the depth information stream will be subjected to packet losses. This simulates a scenario of a congested network whose routers discard some low-priority packets, assuming a simulcast transmission of texture and depth streams having texture packets labeled as more important (high-priority) than depth packets. This is a desirable and quite realistic scheme, in the sense that in case of severe network congestion, clients may still be able to view 2D video with very good quality. Afterwards, further approaches will also include texture packet-losses, as described in section 5.6.

The experiment setup for the first approach is shown in Figure 5.1. It assumes an independent encoding and transmission of texture and depth maps. Some depth packets are dropped by the transmitter-simulator according to the error traces generated by the Gilbert-Elliot model. The uncorrupted texture and corrupted depth stream are then used to synthesize a view with an appropriate baseline (see Annex C the used camera parameter settings for each 3D video). Finally, the PSNR and SSIM of the distorted synthesized view are computed with respect to the reference synthesized view. These objective FR-metrics are later used as ground-truth values for

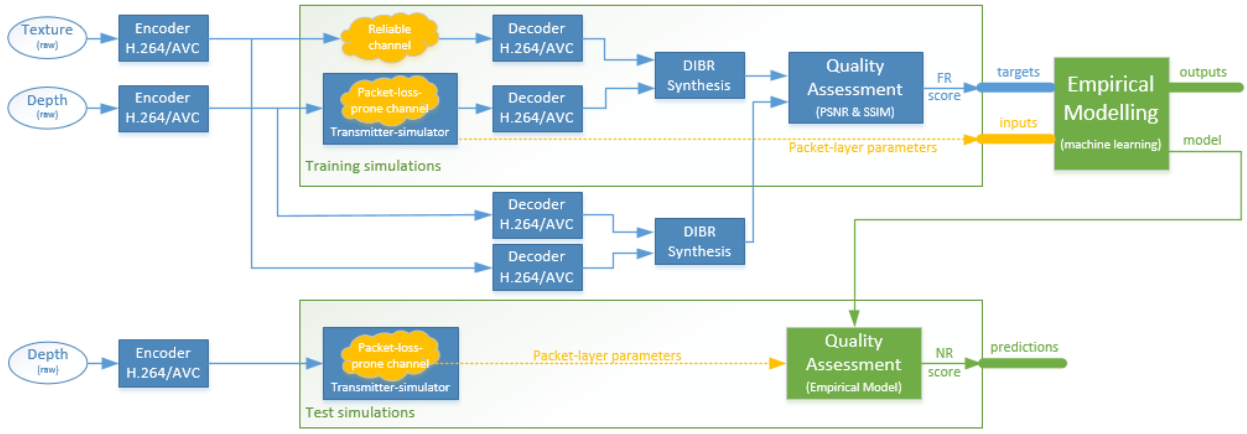


Figure 5.1 – Experiment setup for depth-only loss approach.

the tuning/training of the empirical model; the inputs are a set of parameters obtained from the packets’ headers. In order to obtain an accurate and generalized model, hundreds of simulations for each video are needed. As the decoding and DIBR-Synthesis with VSRS 3.5 are very time-consuming tasks, the use of parallel computing⁶ is imperative. All videos were encoded with the JM version 18.0 (latest release), high profile, and decoded with JM version 15.0, as it is more robust to packet-losses than the latest release.

5.2 Video dataset and encoder setting parameters

The 3D video sequences used in this experiment setup are available online and are entitled: Balloons, Kendo, Champagne Tower and Poznan CarPark. A description of the characteristics of these videos is available in Annex D. All texture and depth videos were encoded with fixed GOP structure, 8 slices (fixed number of macroblocks), inter motion search with reference to two non-B previous frames, motion search range window of 64 pixels, and entropic encoding CABAC. Quality control was set by fixing QP, such that the mean bitrates are compatible with the current transmission bandwidth, and the bit budget division between texture and depth has the recommended ratio listed in Table 5.1.

5.3 Sequence- and GOP-level quality assessment

Full-reference quality assessment target scores of all simulated videos are computed for each i^{th} frame, according to equations (3.3) and (3.6), and then averaged over a GOP of K_G frames or over the entire sequence of K_S frames, according to equations (5.1) and (5.2) :

⁶ Parallel simulations were performed on a Cray clustered computer, owned by Instituto de Telecomunicações – Coimbra

3D Video (spatial resolution)		GOP structure	GOP length (frames)	QP (I / P / B)	Bitrate (kb/s)	Bit ratio (%)	Average PSNR (dB)
Balloons (1024x768)	Tex.	I-B-B-P-B ...	15	28 / 30 / 30	1248	86.4 %	41.55
	Dep.	I-B-B-B-P-B ...	30	36 / 38 / 39	196	13.6 %	39.37
Kendo (1024x768)	Tex.	I-B-B-P-B ...	15	28 / 30 / 30	1245	80 %	42.45
	Dep.	I-B-B-B-P-B ...	30	36 / 38 / 39	300	20 %	38.40
Champagne Tower (1280x960)	Tex.	I-B-B-P-B ...	15	28 / 30 / 30	1303	91 %	41.71
	Dep.	I-B-B-B-P-B ...	30	32 / 34 / 35	129	9 %	45.568
Poznan CarPark (1920x1088)	Tex.	I-B-B-P-B ...	15	28 / 30 / 30	2627	79.8 %	38.171
	Dep.	I-B-B-B-P-B ...	30	30 / 30 / 30	666	20.2 %	36.963

Table 5.1 – Encoder setting parameters of the videos used in the depth-only loss approach.

$$MSE_{SeqGOP} = \frac{1}{K_{SG}} \sum_{i=1}^{K_{SG}} MSE_i \Rightarrow PSNR_{SeqGOP} = 10 \log_{10} \frac{255^2}{MSE_{SeqGOP}} \quad (5.1)$$

$$SSIM_{SeqGOP} = \frac{1}{K_{SG}} \sum_{i=1}^{K_{SG}} SSIM_i \quad (5.2)$$

Note that the PSNR of average MSE is preferred than the direct average of PSNR, because no-impaired frames have infinite PSNR.

5.4 Single-input models

Empirical models need to have at least one input parameter, and as the number of relevant inputs increases, the model accuracy is expected to increase. The first modeling experiments involved a model with a single input parameter: the *PLR* of the sequence (or GOP). The objective is to predict the $PSNR_{Seq}$, $PSNR_{GOP}$, $SSIM_{Seq}$ and $SSIM_{GOP}$ knowing only the *PLR* of the depth stream, which can be computed from the discontinuities of the sequence number present in RTP headers of the depth stream.

The results of sequence-level – resp. GOP-level – mean PSNR and mean SSIM score estimation are shown in the scatter plots of Figure 5.2 – resp. Figure 5.3 –, each point representing the result of one simulation. Plots on the left represent the target (or real) quality scores on y-axis, in terms of *PSNR* and *SSIM*, and the *PLR* on x-axis, as well as the fitting curves, for all simulations of the four videos; plots on the right represent the quality score predicted from the fitting curves of left plots, on y-axis, and the target scores on x-axis. According to the scatter point distribution, for *PSNR* vs. *PLR* fitting curves we chose the power law of equation (5.3), and for *SSIM* vs. *PLR* fitting curves we chose the 1st order polynomial law of equation (5.4).

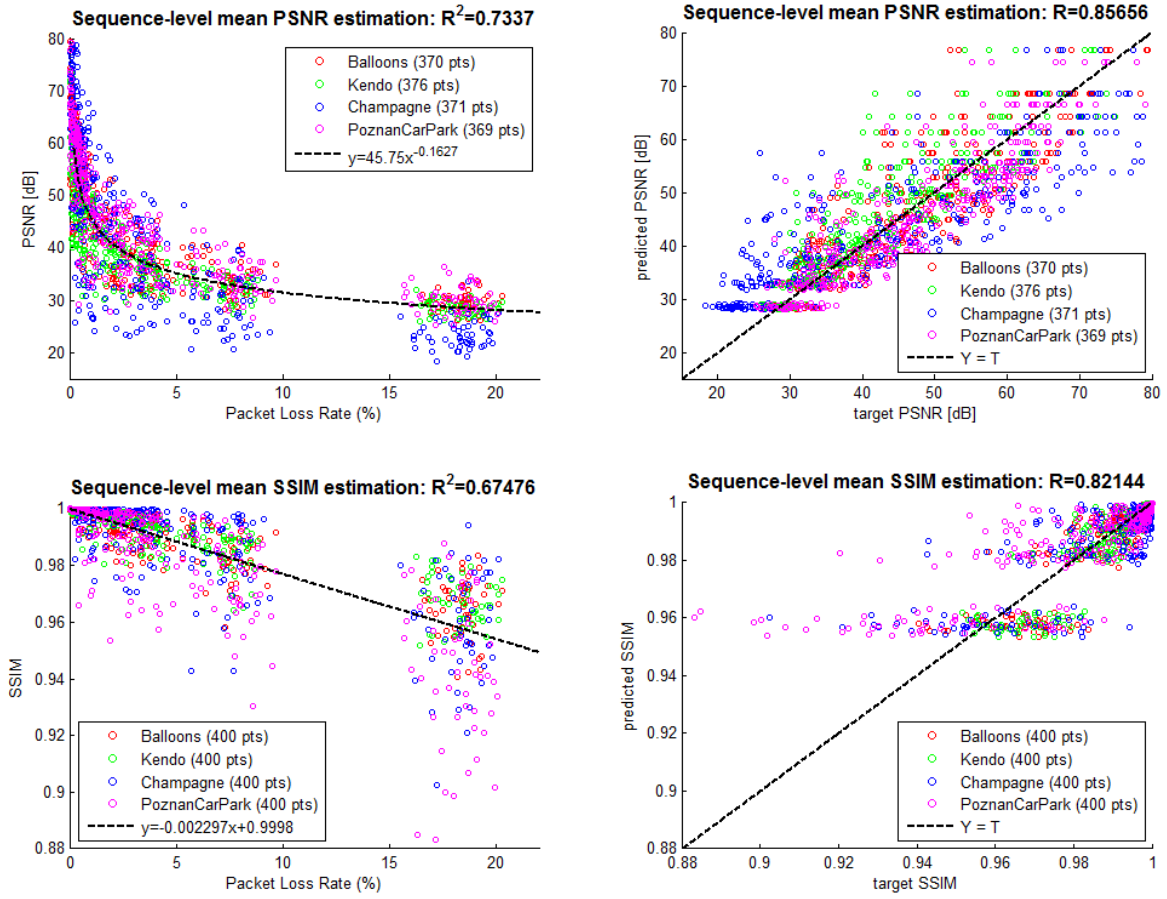


Figure 5.2 – Sequence-level *PSNR* and *SSIM* estimation with a single-input parameter (depth-only loss approach).

$$PSNR = a \cdot PLR^b \quad (5.3)$$

$$SSIM = a \cdot PLR + b \quad (5.4)$$

These models prove to be very inaccurate, as can be observed from the very disperse scatter plots, low Pearson correlation coefficients R , and low coefficients of determination R^2 . The reason for this fact is very simple: not all packets are of equal importance, and so their losses have different impacts on *PSNR* and *SSIM*. As discussed in chapter 4, not all slices are encoded with the same prediction modes, there's a dependency chain between frames introduced by the inter-frame prediction that causes error propagation with more or less severity, and concealment techniques are different for different types of frames. Note how much bigger is the dispersion in GOP-level quality prediction in comparison with the dispersion in sequence-level fashion; the latter one is affected by the averaging effect. Note also that the results for the Champagne Tower video (blue marks) are the most dispersive ones, mainly due to its high-contrast depth maps in comparison with the low-contrast of the other videos' depth maps.

These single-input models have proved to be extremely inaccurate. The next section explores other modeling techniques with multiple inputs, which have much better prediction accuracy.

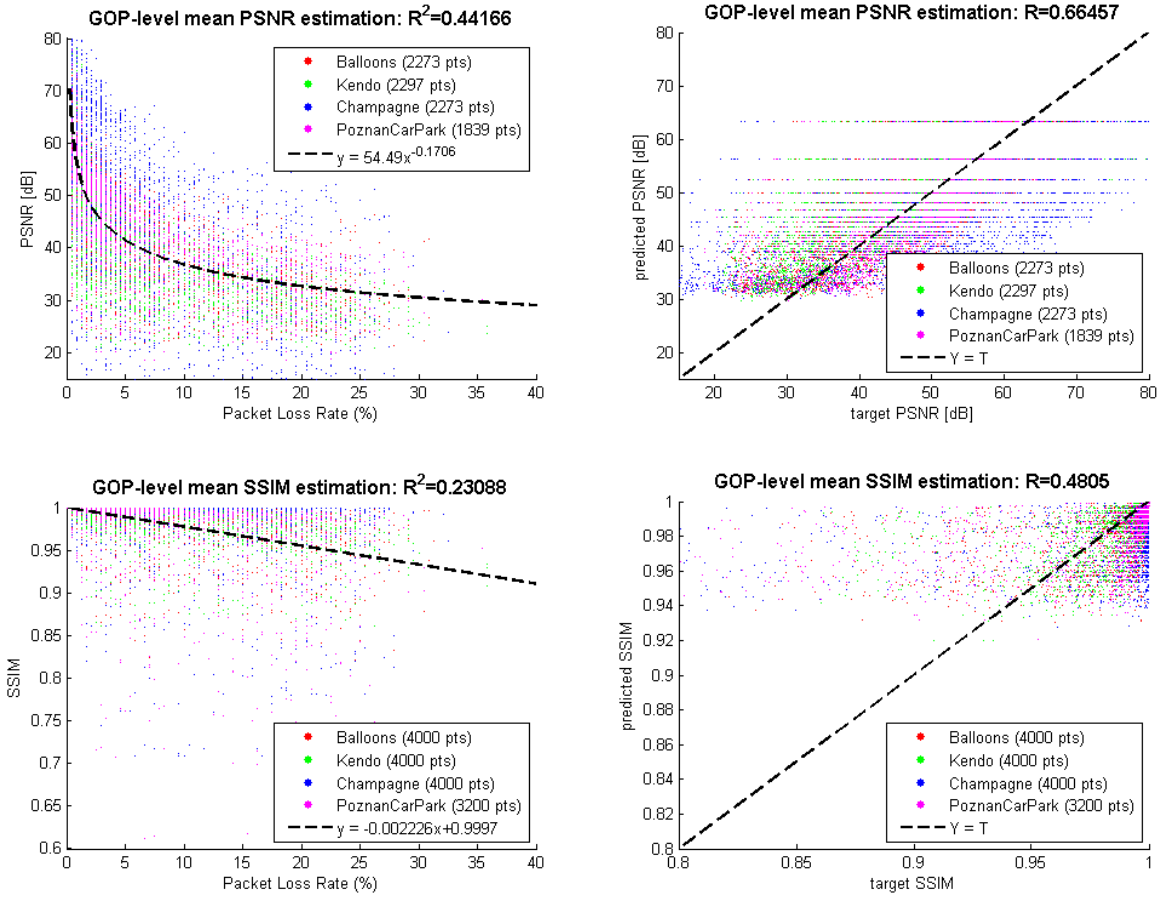


Figure 5.3 – GOP-level *PSNR* and *SSIM* estimation with a single-input parameter (depth-only loss approach).

5.5 Neural networks based models

As the number of inputs to the model increases, it becomes difficult to find regression functions and represent them in a multiple-axis plot. Given its good fitting properties, artificial neural network curve fitting will be adopted from now on, following similar approaches of [71]–[73].

A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons can fit multi-dimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer. In Figure 5.4 is represented a partially detailed two-layer network. It consists in N inputs, H hidden neurons and a single output. The activation functions of the first (hidden) layer and the second (output) layer are resp. the hyperbolic-tangent (sigmoid function) and the identity function. The output is then given by:

$$y(x) = \sum_{j=1}^H \left(w_j^{\text{out}} \cdot \tanh \left(\sum_{i=1}^N w_{ji}^{\text{in}} \cdot x_i + b_j^{\text{in}} \right) \right) + b^{\text{out}} \quad (5.5)$$

where w^{in} and w^{out} are the weights of the first and second layers, b^{in} and b^{out} are the bias of the first and second layers, which are adjusted during in the training phase.

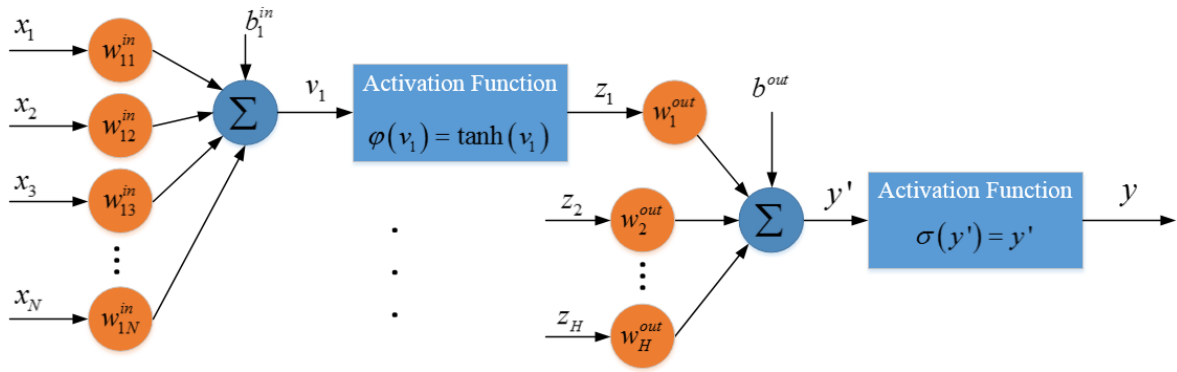


Figure 5.4 – Partially detailed two-layer network corresponding to equation (5.5).

Matlab® *nftool* uses the Levenberg-Marquardt backpropagation algorithm [74] for training the neural network. Given a sufficiently large set of inputs and targets, *nftool* randomly divides them into three groups: the training set used in learning iteration, the validation set used to measure network generalization and to halt training when generalization stops improving, and the test set that provides an independent measure of the network performance during and after training. In all the neural network models trained in this work, the training set is 50% of the overall input/target set, the validation set is 20%, and the test set is 30%. Note, however, that the training algorithm shows a little instability, due to the random initial-values for the variables and random data division, in the sense that different training sessions lead to slightly different results.

As not all packets are of equal importance, it is important to divide the *PLR* into three parameters: the *PLR* of each slice type (I, P or B), defined in the slice header. The size in bytes of the lost packet is also relevant, because the larger the lost packet is, the more impaired the corresponding slice is expected to be, as more information is lost⁷. The size of the lost packet is estimated as the packet size (real or estimated) of the co-located slice of the same type in the previous frame (see Annex E). Thus, adding the estimated sizes of the all three types of lost packets gives us three more parameters. The total size in bytes of the entire sequence, corresponding to the sum of the received packets' size and the lost packets' estimated size may also play a role for accuracy improvement, as it provides an estimate for the lost byte ratio. It gives us three more parameters. Finally, it is important to take into account the temporal location of the losses inside a GOP; for instance, due to error propagation, a lost P-slice affects the entire GOP if it is located at the beginning, but not so much if it is located in the middle or at the end of the GOP. By knowing exactly which packets were lost, it is possible to infer the affected-frame ratio at the packet-layer, in either sequence- or GOP-level.

⁷ Recall that it is assumed transmission over NAL/RTP/UDP/IP protocol, with variable-size packets, each of them containing a single NAL unit corresponding to a single slice, which is different from the fixed-size packet shown in Figure 2.5.

Input	Packet-layer Parameter (depth stream only)	Frame type
x_1	Packet Loss Rate (<i>PLR</i>)	P
x_2		B
x_3		I
x_4	Lost number of Bytes (<i>LB</i>)	P
x_5		B
x_6		I
x_7	Total number of Bytes (<i>TB</i>)	P
x_8		B
x_9		I
x_{10}	Affected-frame Rate (<i>AFR</i>)	all

Table 5.2 – Input parameters for 3D video quality assessment with neural network accurate models.

The use of these input parameters, specified in Table 5.2, assume the GOP length and structure as well as the number of fixed-size slices used are known. If these encoding settings are not known *a priori* or vary in time, they can be extracted at the bitstream-layer during decoding or even at the packet-layer by looking into the headers of a set of consecutive packets. This approach was implemented in a C++ program that is included in Annex F.

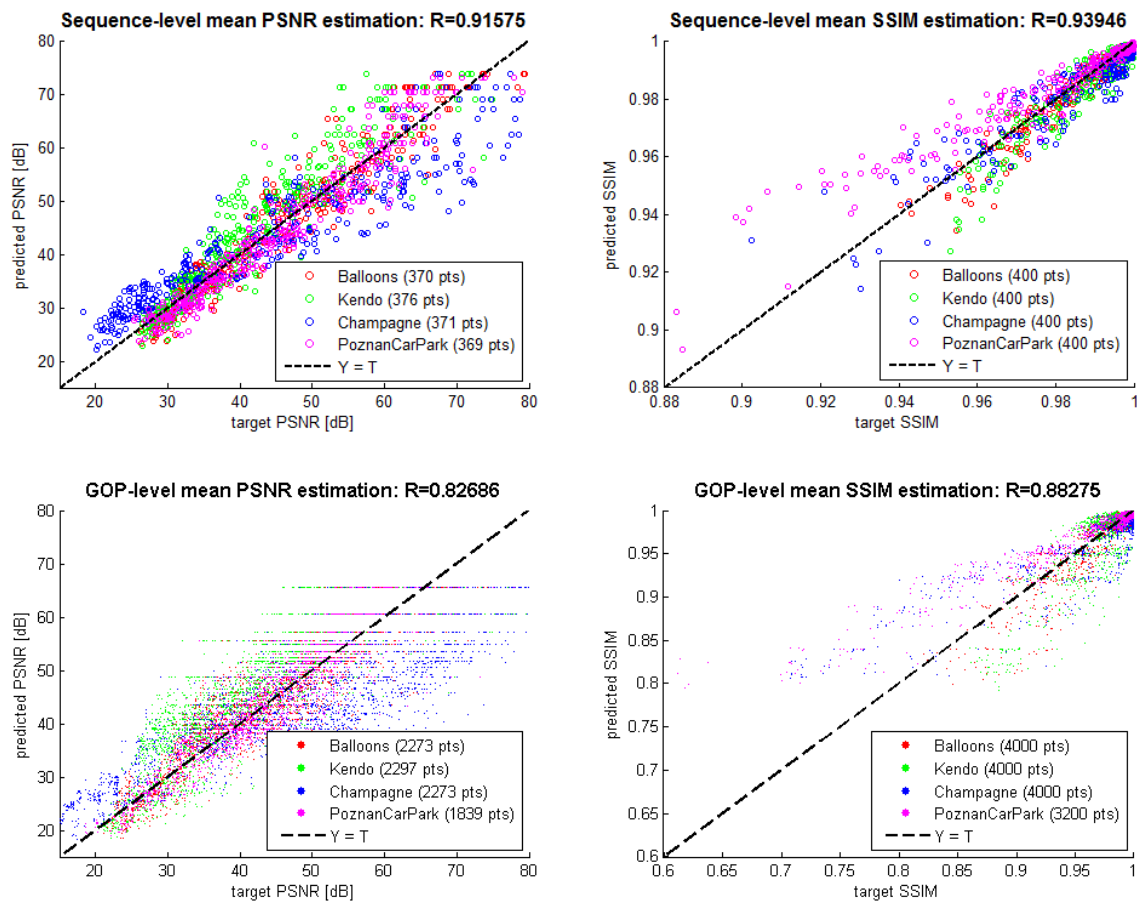


Figure 5.5 – Sequence- and GOP-level *PSNR* and *SSIM* estimation with three input parameters: x_1 , x_2 and x_3 (depth-only loss approach).

In order to show the relative importance of the lost bytes, total bytes and affected-frame rate of the coded depth information stream to model the neural network, Figure 5.5 shows the performance of the network trained with only three input parameters (*PLR* for each type of frame) and 10 hidden nodes. The accuracy of the neural network models to predict the mean *PSNR* and *SSIM* improved significantly when compared with the single input models, as can be shown in Table 5.3. However, they are not as accurate as the neural network models with all 10 input parameters and 20 hidden nodes, shown in Figure 5.6.

These models were trained with the real number of lost bytes. Plots from Figure 5.7 show that the models do not loose accuracy when using the estimated number of lost bytes.

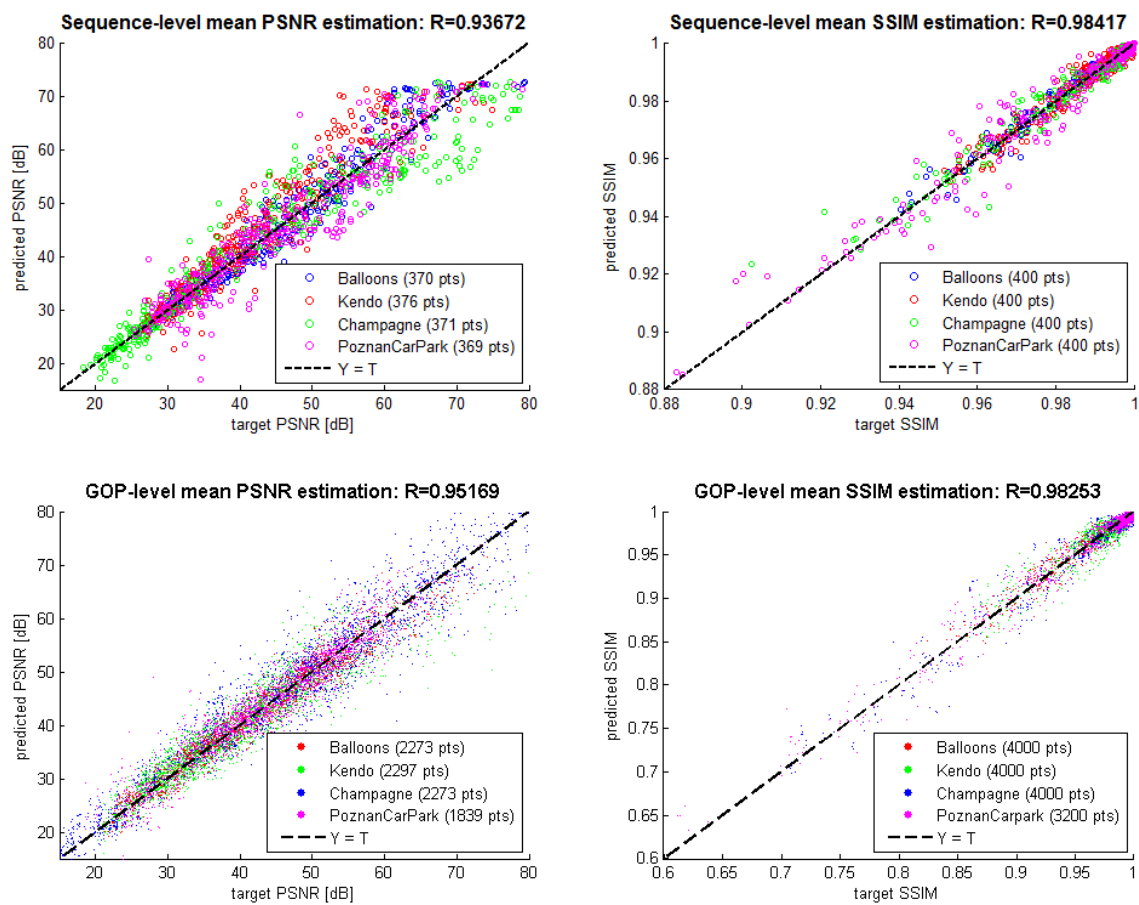


Figure 5.6 – Sequence- and GOP-level PSNR and SSIM estimation with 10 input parameters of Table 5.2 (depth-only loss approach, the number of lost bytes is real)

Number of Inputs	Sequence-level quality prediction		GOP-level quality prediction	
	PSNR	SSIM	PSNR	SSIM
Single input (Figure 5.2 and 5.3)	0.85656	0.82144	0.66457	0.48050
Three inputs (Figure 5.5)	0.91575	0.93946	0.82686	0.88275
Ten inputs (Figure 5.6)	0.93672	0.98417	0.95169	0.98253
Ten inputs (Figure 5.7)	—	—	0.95482	0.98588

Table 5.3 – PLCC of the quality prediction accuracy of the models obtained for depth-only loss approach.

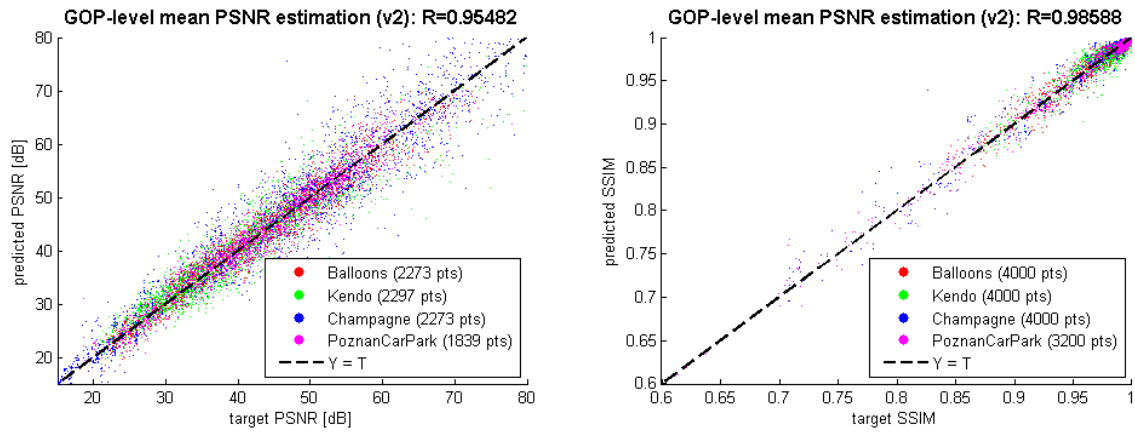


Figure 5.7 – GOP-level PSNR and SSIM estimation with 10 input parameters of Table 5.2 (depth-only loss approach, the number of lost bytes is estimated).

5.6 Modeling the effect of texture losses

So far, we dealt with situations where only depth losses occur; for cases where texture losses also take place, three more experimental setups can be studied. Given the examples of approaches for texture-plus-depth coding and transmission of Figure 2.6, we can obtain quality models for each one. From now on, the scheme of the texture and depth simulcast will be designated by the abbreviation *SIM*, the scheme of frame-compatible texture and depth coding will be designated by *FRM*, and the scheme of scalable texture and depth coding will be designated by *SCA*. Depth-only loss approach scheme will be addressed by *DOL*. The experiment setup of Figure 5.1 is updated into the setup of Figure 5.8 in order to account for texture packet losses.

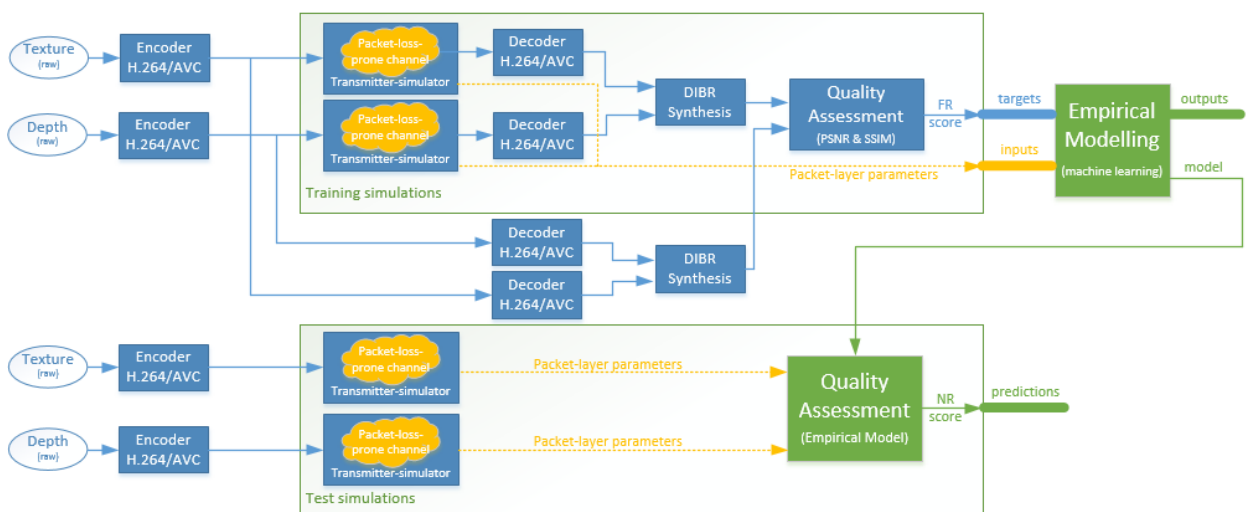


Figure 5.8 – Experiment setup for extended approaches with texture losses (schemes *SIM*, *SCA* and *SIM*).

Let T define the event of loss of (or failure to use) the texture packet of the next slice to be decoded; and let D define the same event related to the depth packet. Figure 5.9 illustrates the probability relationships of these events for each scheme in Venn diagrams.

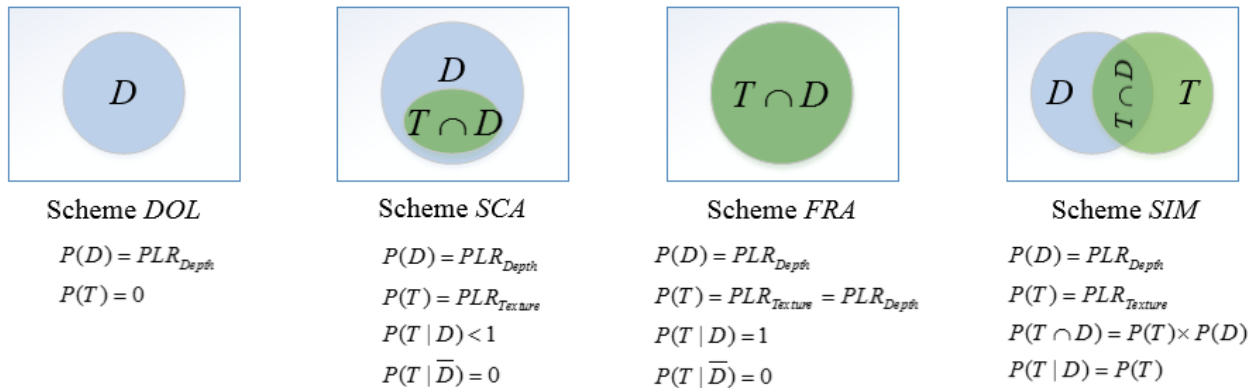


Figure 5.9 – Venn diagrams representing texture and depth packet-losses for four different schemes. T defines the event of a given slice loses (or is not able to use) its texture packet; D defines the event of the same given slice loses (or is not able to use) its depth packet.

In schemes *SIM*, *FRA* and *SCA*, models were obtained with less simulations than in scheme *DOL* because they become much more time-consuming. The videos Champagne Tower and Poznan CarPark used in simulations were replaced by the Lovebird1 and Newspaper. The texture bitstreams were re-encoded with the same GOP-structure as the depth bitstreams, in order to provide full-compatibility with the scheme *FRA*. All texture videos were encoded with the same QP sets, and all depth videos were encoded with the same QP sets, as documented in Table 5.4.

Naturally to account for the texture losses the number of inputs for the modeling of the extended schemes has to be increased. The *PLR*, *LB* and *TB* must be defined to both depth and texture losses independently, for the three types of frames. The *AFR* for schemes *FRA* and *SCA* can be inferred just from the depth stream losses, because affected frames just from texture does not exist in these two schemes. As for scheme *SIM*, whose depth and texture losses can coexist independently, the *AFR* is obtained from the number of frames affected by depth losses, texture losses or both simultaneously.

3D Video (spatial resolution)		GOP structure	GOP length (frames)	QP (I / P / B)	Bitrate (kb/s)	Bit budget (%)	Average PSNR (dB)
Balloons (1024x768)	Tex.	I-B-B-B-P-B ...	30	28 / 30 / 30	1095	84.8 %	41.41
	Dep.			36 / 38 / 39	196	15.2 %	39.37
Kendo (1024x768)	Tex.	I-B-B-B-P-B ...	30	28 / 30 / 30	1173	79.6 %	42.45
	Dep.			36 / 38 / 39	300	20.4 %	38.40
Lovebird1 (1024x768)	Tex.	I-B-B-B-P-B ...	30	28 / 30 / 30	825	89 %	39.17
	Dep.			36 / 38 / 39	102	11 %	42.32
Newspaper (1024x768)	Tex.	I-B-B-B-P-B ...	30	28 / 30 / 30	935	85.5 %	39.28
	Dep.			36 / 38 / 39	159	14.5 %	39.08

Table 5.4 – Encoder setting parameters of the videos used in schemes *SIM*, *FRA* and *SCA*.

Input	Packet-layer Parameter		Frame type	Scheme <i>DOL</i>	Scheme <i>SCA</i>	Scheme <i>FRA</i>	Scheme <i>SIM</i>
x_1	Depth stream	Packet Loss Rate PLR_{Depth}	P	✓	✓	✓	✓
x_2			B	✓	✓	✓	✓
x_3			I	✓	✓	✓	✓
x_4		Lost number of Bytes LB_{Depth}	P	✓	✓	✓	✓
x_5			B	✓	✓	✓	✓
x_6			I	✓	✓	✓	✓
x_7		Total number of Bytes TB_{Depth}	P	✓	✓	✓	✓
x_8			B	✓	✓	✓	✓
x_9			I	✓	✓	✓	✓
x_{10}	Texture stream	Packet Loss Rate $PLR_{Texture}$	P	—	✓	same as x_1	✓
x_{11}			B	—	✓	same as x_2	✓
x_{12}			I	—	✓	same as x_3	✓
x_{13}		Lost number of Bytes $LB_{Texture}$	P	—	✓	✓	✓
x_{14}			B	—	✓	✓	✓
x_{15}			I	—	✓	✓	✓
x_{16}		Total number of Bytes $TB_{Texture}$	P	—	✓	✓	✓
x_{17}			B	—	✓	✓	✓
x_{18}			I	—	✓	✓	✓
x_{19}	Affected-frame Rate (<i>AFR</i>)		all	✓	✓	✓	✓

Table 5.5 – Input parameters for 3D video quality assessment with neural network accurate models (all schemes).

Thus, as specified in Table 5.5, the number of inputs N for schemes *SIM* and *SCA* increases to 19, but for scheme *FRA* it increases to 16, because the *PLR* of the texture stream is the same as of the depth stream. The number of hidden nodes used in neural network training is the same as used in scheme *DOL* ($H = 20$).

For these new simulations, the number of (*PLR*, *MBL*) combinations of packet-loss modeling were restricted to {2%, 8%} for *PLR* and {3, 5} for *MBL*. The error trace files used by texture stream for scheme *SCA* were re-assembled so that:

$$P(T | D) = 0.75 \quad (5.6)$$

Figure 5.10, Figure 5.11 and Figure 5.12 shows the results for sequence- and GOP-level *PSNR* and *SSIM* predictions for resp. schemes *SCA*, *FRM* and *SIM*. We can conclude that the Pearson correlation coefficients of the predicted scores vs. target scores remain very high, as listed in Table 5.6. The computational cost has slightly increased due to new input parameters regarding the texture stream, which are computed with the same algorithms used for depth stream, and the number of hidden nodes used in the neural networks can remain the same.

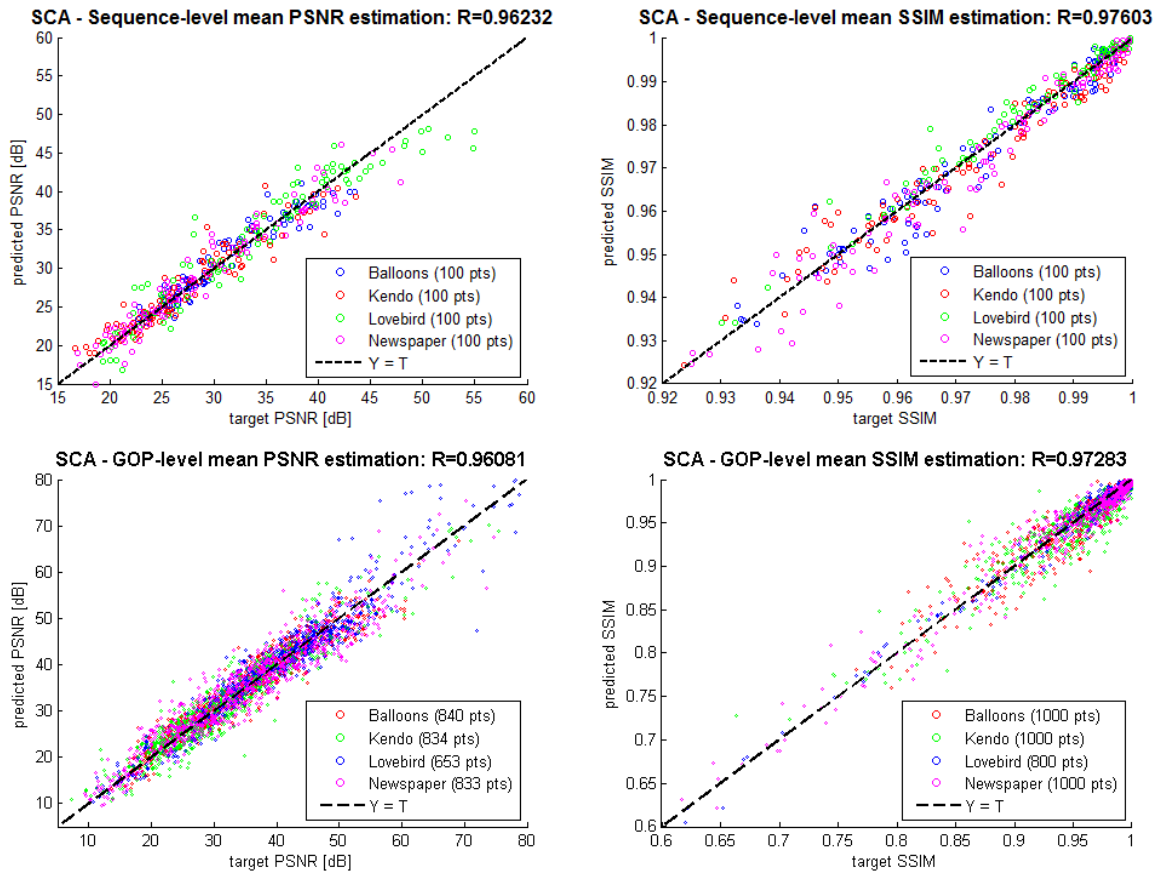


Figure 5.10 – Sequence- and GOP-level PSNR and SSIM estimation for scheme *SCA*, with 19 input parameters specified Table 5.5 (with the real number of lost bytes).

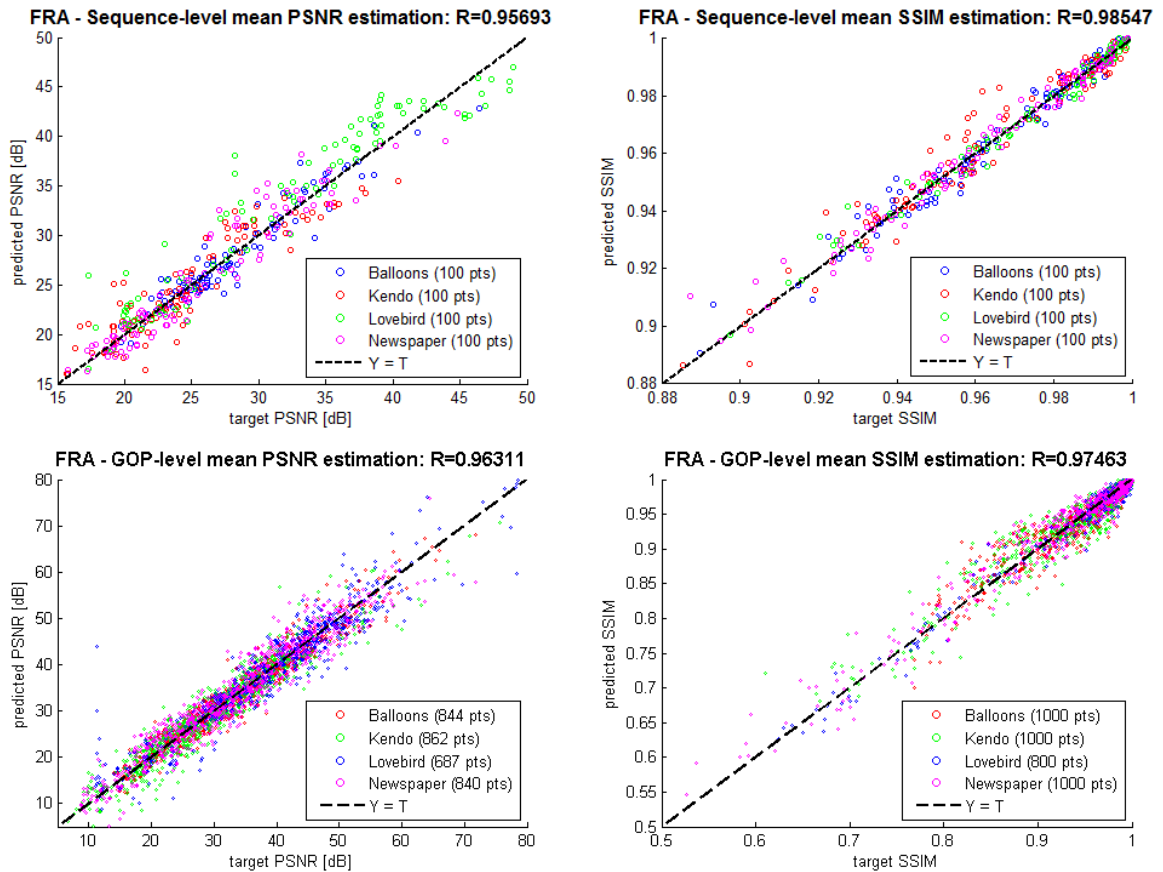


Figure 5.11 – Sequence- and GOP-level PSNR and SSIM estimation for scheme *FRA*, with 16 input parameters specified Table 5.5 (with the real number of lost bytes).

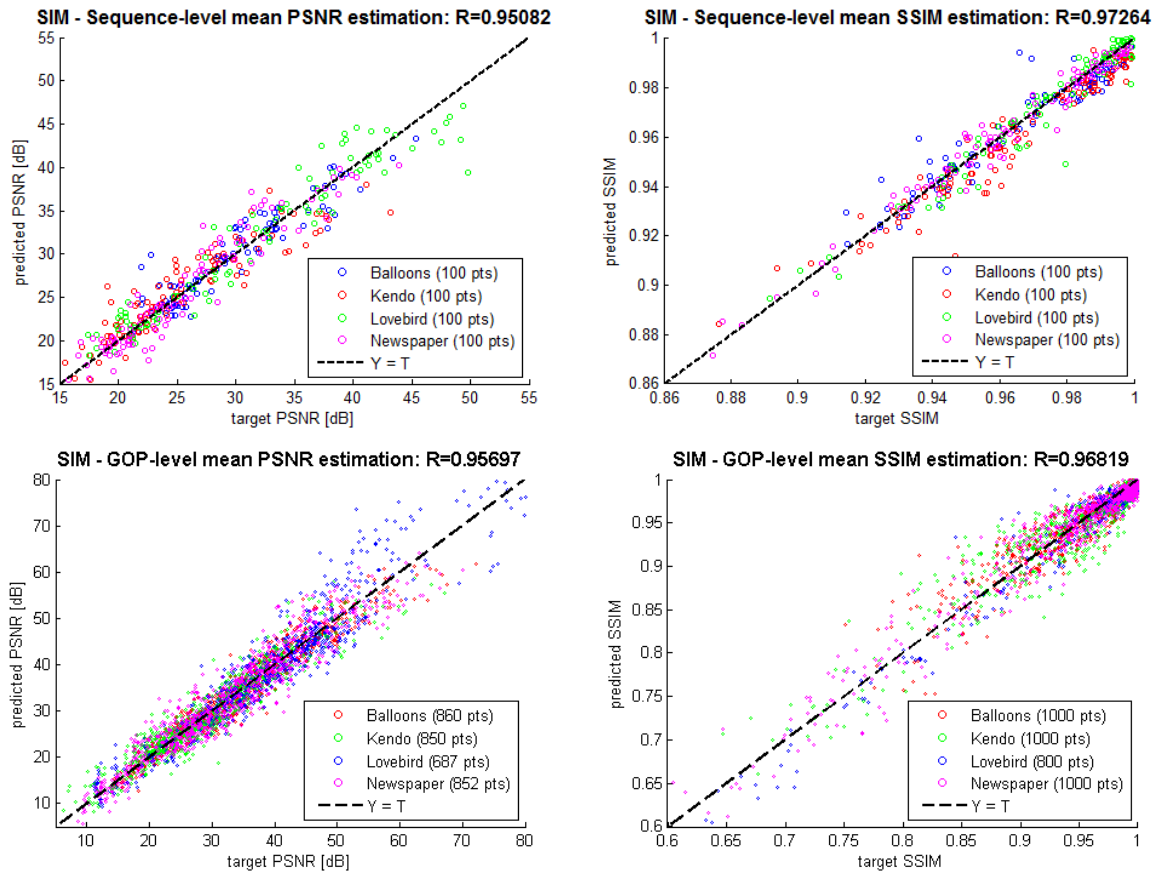


Figure 5.12 – Sequence- and GOP-level PSNR and SSIM estimation for scheme *SIM*, with 19 input parameters specified Table 5.5 (with the real number of lost bytes).

Scheme	Sequence-level quality prediction		GOP-level quality prediction	
	PSNR	SSIM	PSNR	SSIM
<i>DOL</i>	0.93672	0.98417	0.95169	0.98253
<i>SCA</i>	0.96232	0.97603	0.96081	0.97283
<i>FRA</i>	0.95693	0.98547	0.96311	0.97463
<i>SIM</i>	0.95082	0.97264	0.95697	0.96819

Table 5.6 – PLCC of the quality prediction accuracy of the models obtained for all schemes.

In the next chapter, we will confirm that the neural network models used for predicting the sequence-level PSNR and SSIM scores, for scheme *DOL*, are accurate enough for predicting the perceived 3D video quality in terms of a mean opinion score (MOS).

Chapter 6 - Subjective quality assessment of 3D video

6.1 Introduction

In order to measure the correlation between the objective sequence-level quality scores from the models obtained in chapter 5, for scheme *DOL*, and the perceived 3D video quality in terms of a mean opinion score (MOS) which is the *de facto* QoE indicator, a subjective quality assessment study was conducted. There's no optimal methodology to evaluate the quality perceived by humans of 3D video subjected to rare events with unpredictable effects, like bit errors or packet losses. However, it is convenient to follow the rules specified in the recommendations [10], [11] for the environment and test conditions, in order to obtain results that can be compared with other researchers' experiments. In this work, the *Double-Stimulus-Continuous-Quality-Scale* (DSCQS) methodology was used, whose procedures and results are explained in section 6.3.

Another subjective quality assessment was also conducted with a different goal: to measure the extent to which people tolerate losses in depth by comparing a no-loss 2D video with its corresponding depth-impaired 3D video. To achieve this goal, the *2D-3D Pair-Comparison* methodology was used, whose procedures and results are explained in section 6.4.

6.2 Test Environment and subjects

The subjective tests were conducted in a quiet and low-illuminated room. A set of software applications with highly intuitive graphical interfaces were specially designed to be used as the grading and voting consoles on a tablet-PC, as shown in Figure 6.1. A server workstation was used to reproduce all the test sequences in a random order, as well as to register in a database the subjective quality results. A 20-inch Philips WOWvx 9-view autostereoscopic display was used to display the 3D video test sequences, and the viewer was comfortable seated in front of it at the optimal distance of 80 cm. The overall time duration of the evaluation sessions (DSCQS and *Pair-Comparison*) was about 28 minutes. A total of 30 male and 5 female voluntary viewers, aged from 21 to 47, participated in the subjective assessment experiments. Most of the participants were students and only two of them were already familiar with this kind of experiment. All participants have good visual acuity and good stereo vision, as verified with the so called "fly" depth acuity test. Before each evaluation session, the test administrator explained to the participant, in detail, the evaluation process and its objectives to clarify any doubts the participants might have had.

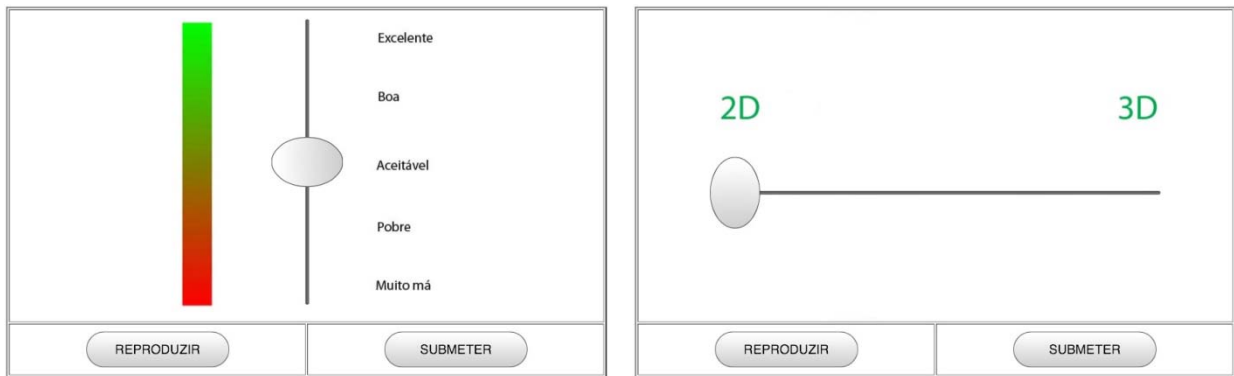


Figure 6.1 – Grading console for DSCQS (left) and voting console for PC (right), written in Portuguese.

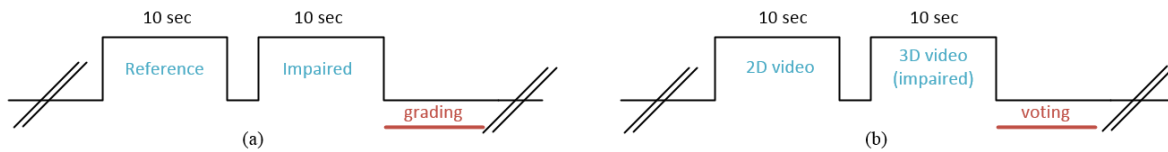


Figure 6.2 – Presentation structures: (a) DSCQS session, (b) 2D-3D Pair-Comparison session.

6.3 DSCQS session: procedures and results

The adopted DSCQS methodology conforms with the ITU-R BT.500-11 [10], but with a minor change. The normative procedure states that the pair reference and distorted video are presented twice, in sequence and in random order such that the viewer assesses both videos, in a continuous quality scale ranging from 0 to 100, but without knowing *a priori* which video is the reference. Then, the differential score is calculated. However, this methodology turns out to be very time-consuming. In order to obtain a larger result set in shorter time, and to prevent eye fatigue from 3D viewing, each pair reference-distorted is presented only one time and the viewer knows *a priori* that the first video to appear is the reference. Thus, the viewers were asked to evaluate the quality of the second video (distorted) with respect to the first video (reference), in the continuous quality grading scale of Figure 6.1 (left) which goes from *very bad* to *excellent*.

The video presentation structure is shown in Figure 6.2 (a). Five impaired versions of each 3D video simulated with scheme *DOL*, with PLR ranging from 0.4% to 20% (see Annex G for further details), were given to the participants for evaluation in a random order. Each distorted video is assigned to a discrete five-rank w according to its PSNR, as explained in Figure 6.3. To reduce the result of inconsistent evaluations each video pair was presented twice, in random moments, but the participants were not aware of this fact. Thus, each individual session collects 40 scores, with 2 scores for each video (a minimum and a maximum, if not equal). A single opinion score per viewer, for each distorted video j , can be obtained with an average of the two scores, weighted according to its rank w , as shown in equation (6.1):

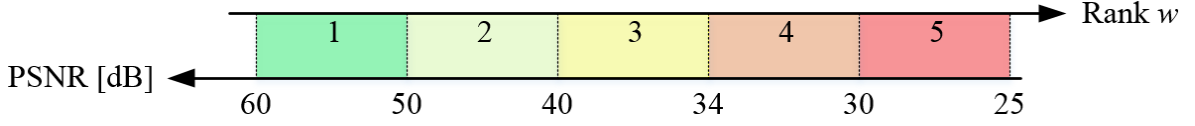


Figure 6.3 – Ranking criteria for distorted 3D videos.

$$Score_{avg,j}(w) = (0.25w - 0.25) \cdot Score_{min,j} + (1.25 - 0.25w) \cdot Score_{max,j} \quad (6.1)$$

In this way, low-loss videos will be given a greater weight to the maximum score, and vice-versa. Then, differential (or delta) scores, for each video j , were obtained as:

$$\Delta Score_{avg,j} = 100 - Score_{avg,j} \quad (6.2)$$

Letting N be the number of viewers, the $DMOS$ for each video j can be calculated as:

$$DMOS_j = \frac{1}{N} \sum_{i=1}^N (\Delta Score_{avg,j})_i \quad (6.3)$$

Figure 6.4 shows the results of the $DMOS$ versus $PSNR_{Seq}$ (left) and $DMOS$ versus $SSIM_{Seq}$ (right) for the 20 evaluated videos, with error bars representing the 95% confidence interval of each $DMOS$, derived from the standard deviation and size of each sample as:

$$\left[DMOS_j - \delta_j, DMOS_j + \delta_j \right], \quad \text{with} \quad \delta_j = \frac{1.96}{\sqrt{N}} \cdot \sqrt{\sum_{i=1}^N \frac{\left(DMOS_j - (\Delta Score_{avg,j})_i \right)^2}{(N-1)}} \quad (6.4)$$

Also plotted are the fitted logistic curves, according to expression (3.2) (and repeated in (6.5)), with fitting coefficients a_1 , a_2 and a_3 given in Table 6.1 and OQM meaning objective quality metric ($PSNR_{Seq}$ or $SSIM_{Seq}$). As we can state by the PLCC, the $DMOS$ is highly correlated with $DMOS$ predicted from $PSNR_{Seq}$ – even better than with $DMOS$ predicted from the $SSIM_{Seq}$ –, which demonstrates the assumption made on chapter 3.2: that the PSNR is a good predictor of subjective video quality, for specific impairments due to packet-loss events.

$$DMOS_{Predicted} = \frac{a_1}{1 + e^{a_2(OQM + a_3)}} \quad , \quad a_1, a_2, a_3 \text{ are fitting coefficients} \quad (6.5)$$

	a_1 ⁸	a_2	a_3	R^2	PLCC between $DMOS$ and $DMOS_{Predicted}$
$DMOS$ vs. $PSNR_{Seq}$	100	0.1755	-36.63	0.95998	0.97978
$DMOS$ vs. $SSIM_{Seq}$	100	126	-0.9891	0.78749	0.88741

Table 6.1 – Logistic fitting coefficients and PLCC of the plots of Figure 6.4.

⁸ The value of this coefficient was forced to 100 because it is the maximum value of the DSCQS adopted scale.

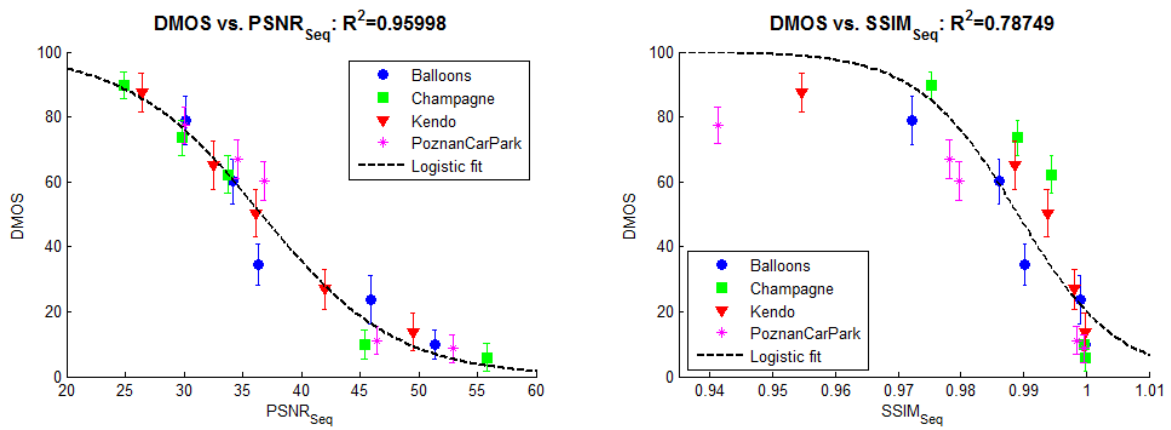


Figure 6.4 – DMOS vs. $PSNR_{Seq}$ (left) and DMOS vs. $SSIM_{Seq}$ (right) for the 20 evaluated videos.

6.4 2D-3D Pair-Comparison session: procedures and results

In this session, 30 viewers were presented sequence pairs of 2D video followed by an impaired 3D video version, as depicted in Figure 6.2 (b), and were asked to indicate (by voting) which one they preferred to view – no middle-choices allowed –, according to the voting console of Figure 6.1 (right). The same impaired videos used in DSCQS are used, but only those ranking from 2 to 5 are presented for voting, just once, resulting in 16 videos and 16 votes per individual session.

The results of this subjective experiment are shown in Figure 6.5. As expected, as the magnitude of the degradation of the depth information increases (due to higher PLR), viewers tend to prefer the “clean” 2D version. Note, however, that not everyone prefers to view an almost “clean” 3D version over the 2D version. The technology of the autostereoscopic 3D display used in these subjective experiments plays an important role in the perceived quality of 3D video, namely the presence of crosstalk between views when the disparity is very large (as in case of Champagne Tower) [75], an effect that may explain the preference of some users for the 2D video over the 3D video.

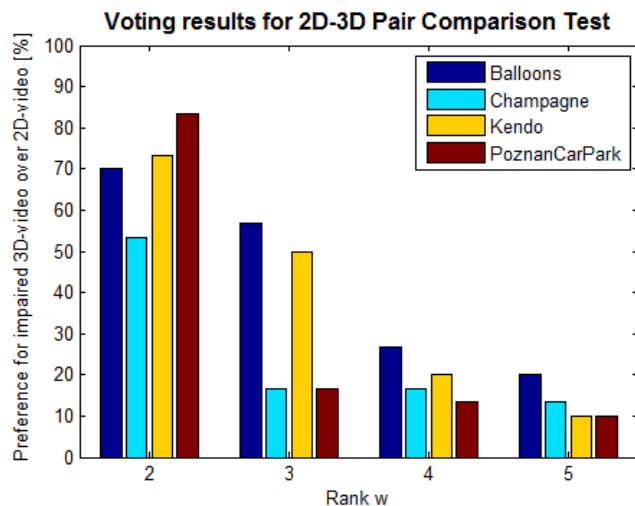


Figure 6.5 – Voting results for 2D-3D Pair Comparison Test.

Chapter 7 - Empirical hybrid models for frame-level synthesized view quality assessment

7.1 Context, objective and procedures

This last chapter presents some low-complexity techniques for empirical modeling of the synthesized 3D video quality, in a frame-wise fashion. The objective of these techniques is to estimate an objective quality score (e.g. SSIM) of a synthesized frame/slice, based on a set of media-layer parameters. The techniques introduced take into account the fact that the frame/slice which quality is to be estimated has been impaired either due to the loss of its own texture or depth packets and also, possibly, due to error propagation. While in this chapter we are not taking into account network statistics like PLR or MBL, studied in chapter 5, it is still convenient to look into the headers of the packets and to have some knowledge about the GOP and slice structure of the texture and depth streams, in order to infer which slices have been actually damaged by packet losses. Therefore we cannot classify this kind of models as pure media-layer: they are hybrid models (Table 3.1) combining packet layer and media layer information. They are well suited for deployment in set-top boxes, because they require the decoded video which is readily available in the decoding terminals; however, for this same reason (access to decoded video) and contrary to the kind of models studied in chapter 5, these models are not efficient to be implemented in a network node.

The four encoding and transmission schemes considered in chapter 5 will be considered in this study as well. The same procedures for inferring which slices are affected by self-loss or error propagation in packet-layer models will be used. When checking frame-by-frame, following the display order, which slices may be affected, it is convenient to assign them to one of two categories: (a) slices whose co-located slice in the previous frame inside the GOP has not been affected, and (b) slices whose co-located slice in the previous frame inside the GOP has been affected as well. We can define the first group as the *initial-loss slices*, and the second group as the *non-initial-loss slices*, as exemplified in Figure 7.1 which applies the same packet loss pattern of Figure 4.4. Lost I- or IDR-slices are always classified as *initial-loss slices*; due to the error concealment technique adopted by the JM decoder, they are also classified as *intra losses*.

In Figure 7.1, green slices represent no-affected slices; red slices represent *initial-loss* assigned slices; and yellow slices represent *non-initial-loss* assigned slices. Note that other packet-loss patterns would have produced the same slice assignment; in other words, given a specific slice assignment, it is not possible to infer the exact packet-loss pattern (but that is not important).

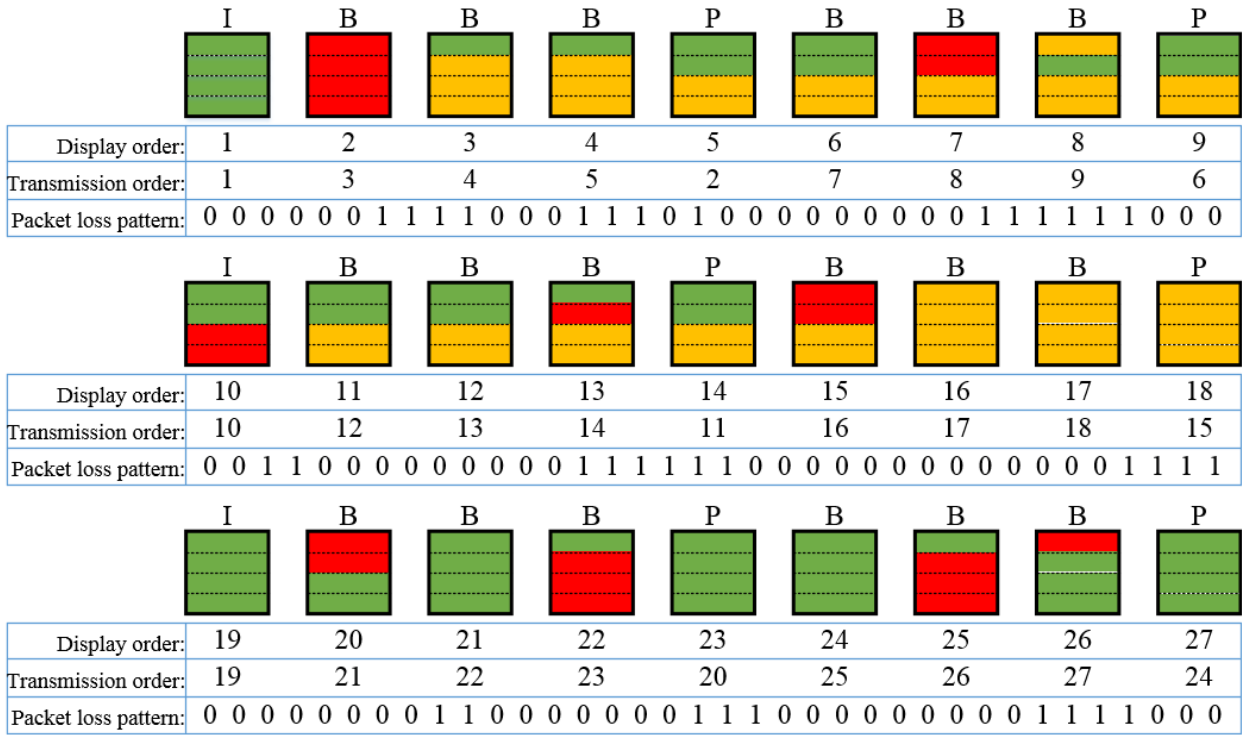


Figure 7.1 – Example of initial-loss classification given the same error pattern as Figure 4.4.

In the simulation setup, once again the (un)corrupted texture and corrupted depth stream are used to synthesize a view. The SSIM of each slice of the distorted synthesized view is computed with respect to the reference synthesized view, and then is used as target of the empirical model. Two approaches will be considered: the first one is to train a single neural network for all affected slices (section 7.4); the second one is to train a neural network for each of the two slice categories (section 7.5). As in chapter 5, hundreds of simulations have been performed to gather data to be used in tuning the quality estimators. The videos used are Balloons, Lovebird1 and Newspaper, with encoding configurations listed in Table 5.4.

7.2 Input packet-layer binary parameters

There are eight binary parameters that can be defined at the packet-layer⁹, to classify the rendered slice: four related to the texture information and four related to the depth information, as shown in Table 7.1. x_{13} and x_{26} identify if the slice has been somehow impaired, either by self-loss or propagated loss of the texture stream or depth stream. As the rendered slices to be assessed are known *a priori* to be impaired, x_{26} is not explicitly used in schemes *DOL*, *SCA* and

⁹ Further research is needed to understand if it is possible to aggregate these eight binary parameters into two quaternary parameters in order to reduce the amount of inputs and to reduce the neural network complexity, but without losing accuracy.

FRA; x_{13} is used only in schemes *FRA* and *SIM* because these are the schemes where texture loss can coexist not simultaneously with depth losses. x_{10} and x_{23} identify if the slices have been (also) impaired by the loss of its own texture or depth packet; they form a unique parameter for scheme *FRA* as the losses are mutual. x_{11} and x_{24} identify if the impaired slice is an I- or IDR-slice, relevant information due to the nature of the error concealment; these two form a unique parameter for scheme *FRA*. Finally, x_{12} and x_{25} identify the assigned category: *initial-loss* or *non-initial-loss*. x_{12} is irrelevant in schemes *DOL* and *SCA* (so it is not used); x_{12} and x_{25} form a unique parameter for schemes *FRA* and *SIM*.

		Texture				Depth							
		D	S	F	S	D	S	F	S				
		O	V	R	I	O	V	R	I				
		L	C	A	M	L	C	A	M				
Media-layer Descriptors	Spatial Complexity (Histogram)	x_1	Mean (μ)	✓	✓	✓	✓	x_{14}	Mean (μ)	✓	✓	✓	✓
		x_2	Standard Deviation (σ)	✓	✓	✓	✓	x_{15}	Standard Deviation (σ)	✓	✓	✓	✓
		x_3	$\sigma(\text{low resolution}) = \beta$	✓	✓	✓	✓	x_{16}	$\sigma(\text{low resolution}) = \beta$	✓	✓	✓	✓
		x_4	Entropy (H)	✓	✓	✓	✓	x_{17}	Entropy (H)	✓	✓	✓	✓
	Temporal Complexity (Differential)	x_5	$\mu(\Delta\text{slice})$	✓	✓	✓	✓	x_{18}	$\mu - \mu(\text{slice}_{\text{PreviousFrame}})$	✓	✓	✓	✓
		x_6	$\sigma(\Delta\text{slice})$	✓	✓	✓	✓	x_{19}	$\sigma - \sigma(\text{slice}_{\text{PreviousFrame}})$	✓	✓	✓	✓
		x_7	$\beta(\Delta\text{slice})$	✓	✓	✓	✓	x_{20}	$\beta - \beta(\text{slice}_{\text{PreviousFrame}})$	✓	✓	✓	✓
		x_8	$H(\Delta\text{slice})$	✓	✓	✓	✓	x_{21}	$H - H(\text{slice}_{\text{PreviousFrame}})$	✓	✓	✓	✓
	Artifacts	x_9	Block discontinuities (η)	–	✓	✓	✓	x_{22}	Block discontinuities (η)	✓	✓	✓	✓
Packet-Layer binary parameters	x_{10}	Self-loss	–	✓	*	✓	x_{23}	Self-loss	✓	✓	*	✓	
	x_{11}	I- or IDR-slice	–	✓	*	✓	x_{24}	I- or IDR-slice	✓	✓	*	✓	
	x_{12}	Initial-loss slice	–	–	*	*	x_{25}	Initial-loss slice	✓	✓	*	*	
	x_{13}	Impaired slice (general case)	–	✓	–	✓	x_{26}	Impaired slice (general case)	<i>i</i>	<i>i</i>	<i>i</i>	✓	
Feedback	x_{27}	Quality output from the previous co-located slice (either been affected or not).							✓	✓	✓	✓	

Table 7.1 – Input parameters for slice-wise quality assessment of the synthesized view with neural network models, and their use in all transmission schemes. The ✓ means that variable are explicitly used as neural network input; the * means that variables of texture and depth are shared; the *i* means that variable are not explicitly used as a neural network input, but as an implicit system control variable; and the – means that variable is not used at all.

7.3 Input media-layer histogram-based descriptors

As listed in Table 7.1, there are three histogram-based low-complexity operators that can be applied to the luminance component¹⁰ of texture and depth slices to compute a series of

¹⁰ Recall from section 3.2 that only the luminance component is used to compute PSNR and SSIM scores. Thus, it is irrelevant to apply these operators on the chrominance components.

descriptors: mean, standard deviation and entropy. They are able to measure, to some extent, the spatial complexity of a slice. The larger the spatial complexity, the more impaired the slice is expected to be if the corresponding packets are lost. In order to measure the temporal complexity of a slice, the same three operators can also be applied to the absolute pixel-differences between the current slice and the co-located slice from the previous frame, referred here as $\Delta slice$ that defined as:

$$\Delta slice = |slice_{CurrentFrame} - slice_{PreviousFrame}| \quad (7.1)$$

Given a slice histogram where l is the gray level (out of a total of L possible values – typically 256), whose probability of occurrence is $p(l)$, we can define the mean, standard deviation and entropy in equations (7.2), (7.3) and (7.4), respectively:

$$\mu = \sum_{l=0}^{255} l \cdot p(l) \quad (7.2)$$

$$\sigma = \sqrt{\sum_{l=0}^{255} (l - \mu)^2 p(l)} \quad (7.3)$$

$$H = -\sum_{l=0}^{255} p(l) \log_2 [p(l)] \quad (7.4)$$

It was defined another operator, referred in Table 7.1 as β , that results from the combination of (7.2) and (7.3): first the slice is partitioned into N macroblocks of 16x16 pixels, then each block is applied the mean operator, and finally the standard deviation is computed for the resulting value set. The result is the same as applying the operator (7.3) to a “raw” low-resolution version (16:1) of the slice, and as with the previous three operators, this one can be used in both spatial and temporal complexity measurement.

Typical artifacts seen on impaired slices due to error propagation are blocking effects, characterized by abrupt discontinuities at the macroblock and slice boundaries. They are indicated in Table 7.1 as η . In Annex H is included the script used to compute the η of a slice.

These media-layer descriptors don't have all the same relevancy. The most important ones are the entropy and the block discontinuities η . Some descriptors are more relevant in the spatial domain than in the temporal domain, and vice-versa. However, as the video content diversity broadens the field of impairment possibilities, it was decided to train the neural networks and to present the results of the slice SSIM prediction using all the descriptors.

7.4 3D-VQM Architecture 1: single neural network model

The overall structure of the hybrid 3D Video Quality Monitor (Architecture 1) is shown in Figure 7.2. The packet-layer and media-layer inputs to the neural network were explained in sections 7.2 and 7.3, but another input – probably the most important one – is used: the SSIM score from the previous co-located slice, either it has been impaired or not.

This feedback input is very important for accurate *non-initial-loss* slice quality assessment, because of error propagation. Thus, we can define a “SSIM propagation”, which fluctuations are due to the remaining input parameters. However, the accuracy of the *initial-loss* slice quality scores may fluctuate, as the previous co-located slice has not been affected, has maximum quality, and thus the score is not correlated at all. In these cases, the packet-layer parameters and media-layer descriptors are the only ones which can be used to predict the quality of *non-initial-loss* slices.

The scatter plots of Figure 7.3, for each considered scheme, show the excellent accuracy (PLCC over 0.99) of the trained neural network models, with 25 hidden nodes. Each point of the plots represents one affected slice SSIM score. Note that the majority of the points, with very accurate prediction along the $Y=T$ line, are from *non-initial-loss* slices. However, these models have considered the real value of the previous co-located SSIM in the training process. If the *initial-loss* slices, in general, do not show very high prediction accuracy, the overall performance of the model is lower than expected (see Figure A.3 of Annex I). Thus, it is very important to ensure that the *initial-loss* slice scores are predicted with good accuracy. The next section explains an extended version of the hybrid 3D Video Quality Monitor with this aim.

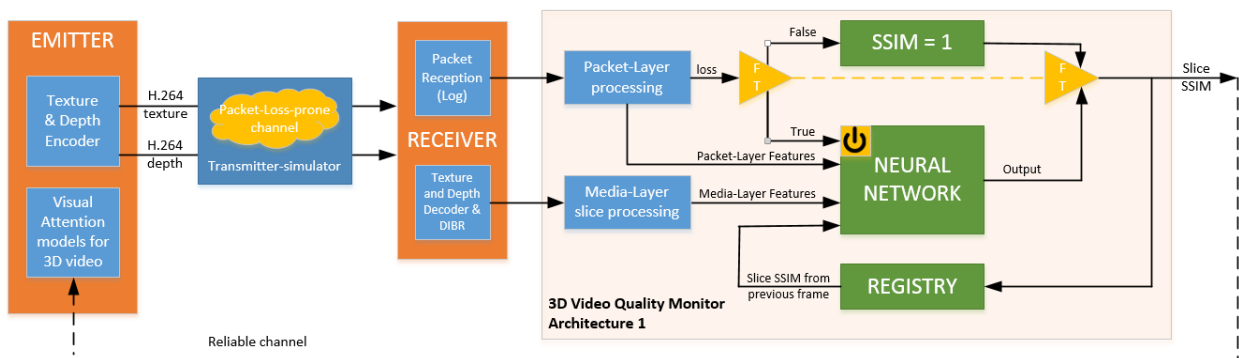


Figure 7.2 – Overall structure of the hybrid 3D Video Quality Monitor (3D-VQM Architecture 1).

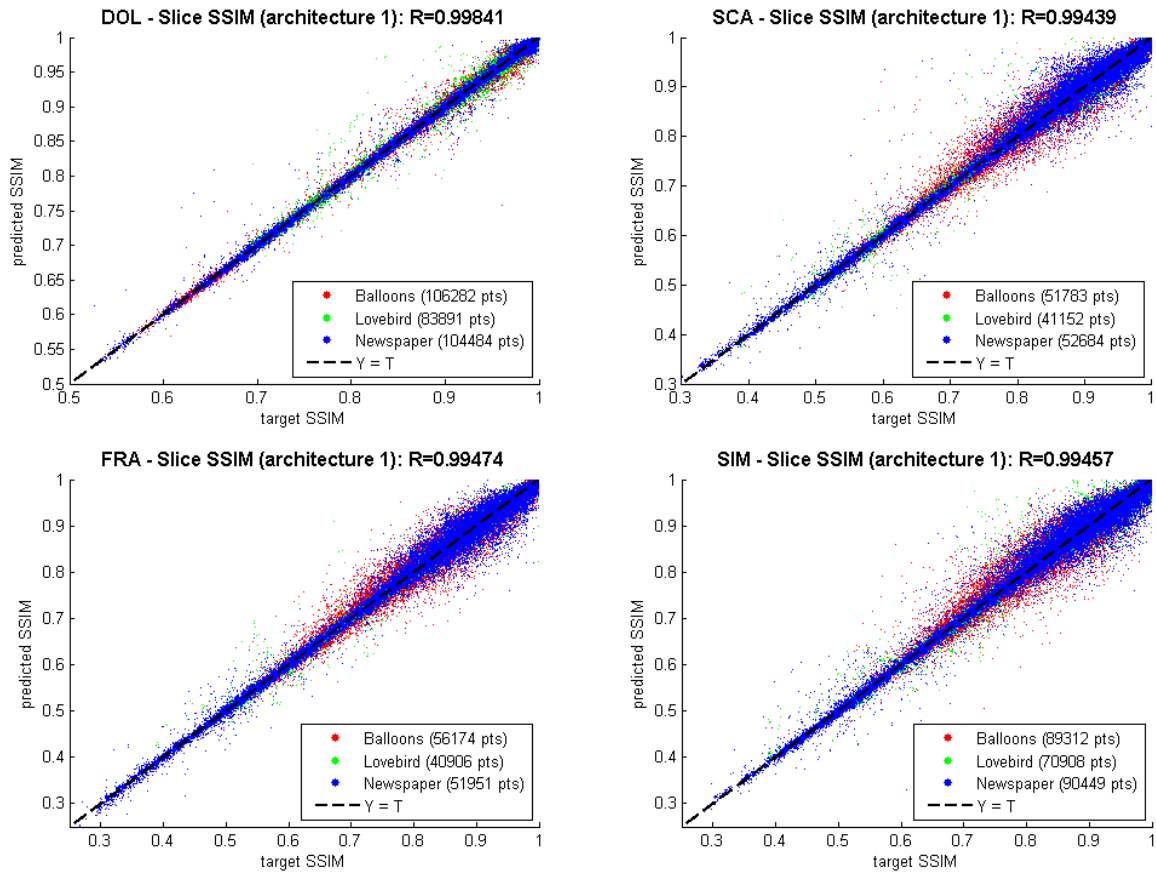


Figure 7.3 – Target slice SSIM versus predicted slice SSIM with the 3D-VQM Architecture 1, for all schemes.

7.5 3D-VQM Architecture 2: double neural network model

In order to decouple the effects of the feedback input in the quality prediction of *initial-loss* slices, architecture 2 was developed with two neural network models: one oriented to *non-initial-loss* slices and another oriented to *initial-loss* slices, this one with no feedback input. Figure 7.4 shows the overall structure of this extended architecture.

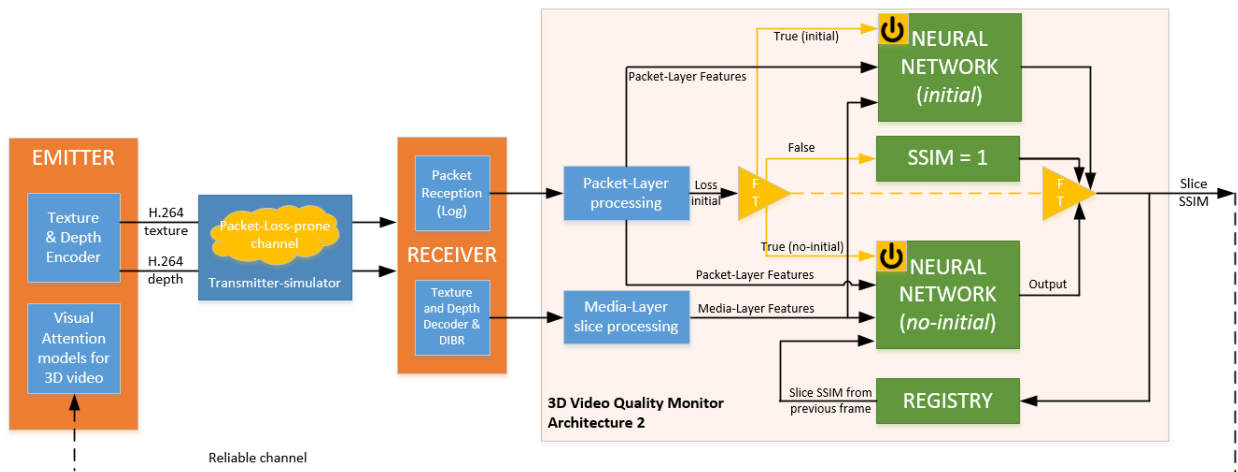


Figure 7.4 – Overall structure of the hybrid 3D Video Quality Monitor (3D-VQM Architecture 2).

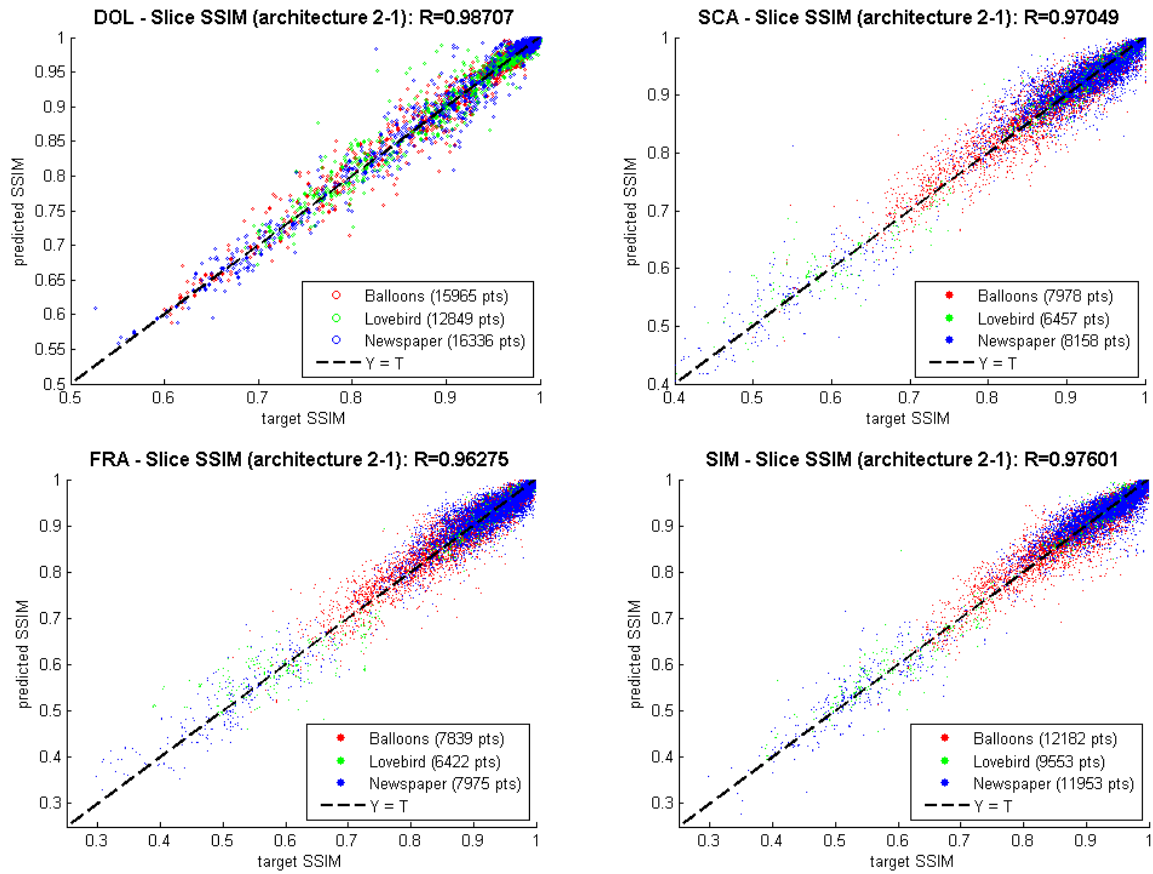


Figure 7.5 – Target slice SSIM versus predicted slice SSIM with the 3D-VQM Architecture 2, for all schemes (neural network dedicated to *initial-loss* slices)

The scatter plots of Figure 7.5, for each considered scheme, show that the accuracy for *initial-loss* slice quality prediction scores is still high (PLCC over 0.96). Thus, the overall performance of the system is expected not to degrade so much due to the “predicted SSIM propagation”. On the other hand, the scatter plots of Figure 7.6 are related to *non-initial-loss* slice quality prediction. As expected from the use of the feedback SSIM, the overall accuracy of these slices’ quality prediction is much higher than the *initial-loss* slice quality prediction (PLCC over 0.99). Figure A.4 of Annex I shows an example of the quality monitor performance with the trained networks for scheme *DOL*, in which the predicted SSIM value is fed back to the *non-initial-loss* neural network input.

7.6 Averaging scores with visual attention models

In an industrial setting, the slice quality scores obtained using the estimators described in the previous chapters and sections can be transmitted to the service or content provider through a reliable transport protocol (e.g. TCP/IP) and integrated in the network’s operation and management framework.

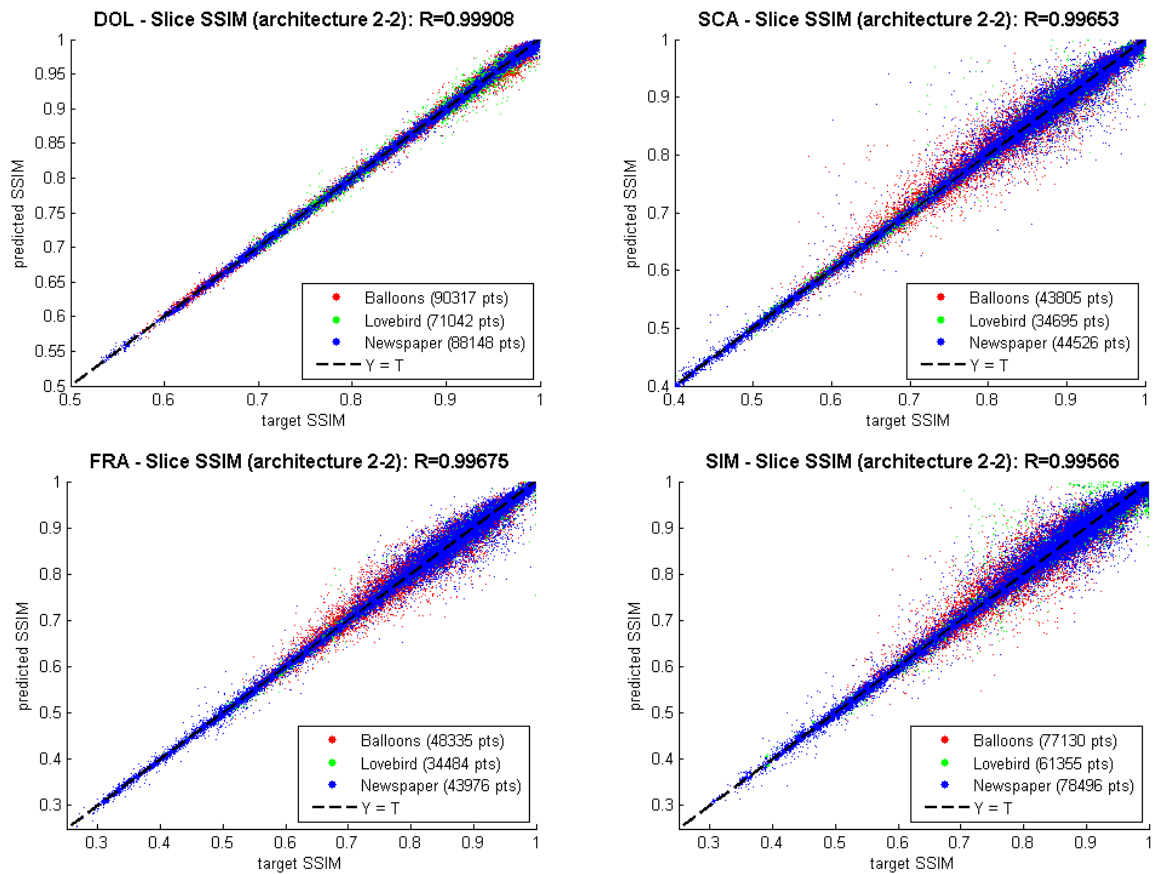


Figure 7.6 – Target slice SSIM versus predicted slice SSIM with the 3D-VQM Architecture 2, for all schemes (neural network dedicated to *non-initial-loss* slices)

If the goal is to obtain an overall score for the entire frame, the simplest solution is to average the scores of the slices that make up that frame and transmit individual frame scores. Conversely if longer term quality indicators are preferable, average scores for a sequence of frames within a (possibly sliding) temporal window can be computed and sent to the quality control system.

However, sometimes this averaging process can produce values that are not very well matched to the human observer perceived quality. To alleviate this problem, visual attention models for multiview 3D video can be used to define weighting factors to incorporate in averaging procedures, so that the average scores are closer to the subjective human observer perceived quality. However interesting, this topic (3D video attention models) is still in the first stages of research, and further work on it is far from the scope of this thesis. As argued by the authors of [76], a simple extension of 2D video visual attention models (like the one described in [77]) to stereoscopic or multiview 3D video doesn't seem to be biologically plausible, due to, for instance, masking effects between views, occlusions or the effect of large disparities. Research on the inclusion of depth information in still 3D image visual saliency has been published very recently [78]. However, it is needed further extensive research concerning human behavioral responses in the visual exploration of 3D video content.

Chapter 8 - Conclusion

In this thesis, the subject addressed was the objective quality evaluation of 3D video transmitted over packet-loss-prone channels. The four objectives proposed in section 1.2 have all been successfully achieved. The empirical quality models discussed and developed in this work yield very good results (which are object of future publication in scientific articles), regarding the assumptions made and methodologies followed – the encoding and transmission setup and the definition of the used input parameters and targets. As discussed in chapter 3, it is very difficult to compare different quality assessment methods developed and published in the scientific literature, because almost all of them have different kinds of applications, different goals and even different assumptions regarding, for instance, the encoding and transmission setup. Nevertheless, this premise deserves a little more discussion, before closing this essay.

The first approach to address the subject of this thesis was to try to assemble no-reference pure media-layer models for 3D video quality assessment, either found in the scientific literature, or proposing new ones. The quality impairment source was already restricted to packet losses, thus a simulation environment was needed to be established: (a) to encode raw texture-plus-depth videos, (b) to packetize their bitstreams, (c) to simulate packet losses, and finally (d) to evaluate the final quality. The first issue to overcome was to decide the digital encoding format to be adopted, how to packetize, and how to discard some packets, in order to decode a video impaired by packet-losses. Ideally, a pure media-layer general-purpose and no-reference quality model does not give any importance to the simulation environment adopted, so any encoding parameters can be used. Soon it became obvious that, currently, there are no such models in the scientific literature. Let's think over it: is it possible to objectively evaluate the quality of a 3D video – or even a 2D video or an image – without any reference or trained model, based only on pixel-domain information? We can think even further: is it possible to subjectively evaluate a short-duration video, or to continuously evaluate a long-duration video, without having some common-sense references regarding to how an “excellent quality” video should look like? Furthermore, the “excellent quality” references are continuously changing due to the advances of technology: a low-resolution video displayed in a large television is typically not assessed as “good quality” anymore, but perhaps two decades ago it was. That's why the subjects who participated in the subjective test described in chapter 6 were asked to evaluate the impaired videos in a comparison basis. Some of them claimed that even the unimpaired videos used as references have very poor quality, not directly related to compression artifacts, but mostly related to the displaying conditions of the used autostereoscopic display, which were far from perfect.

In this way, methods based on only media-layer information may be highly susceptible to erroneous QoE predictions, for instance due to the fact that the original content might be already of *poor* quality (e.g. a very old movie or a video captured with an uncalibrated camera), even if it is transmitted with excellent QoS (i.e. without transmission losses or bit errors). In the thesis author's opinion, media-layer models could benefit from other layers' information (if available), such as parameters extracted from the bitstream or packet-layer, in order to output quality scores consistent with the potential transmission problems. This approach brings more realism to the scope of loss-prone transmission schemes, in the way that media-layer quality assessment of video frames, which are known *a priori* to be impaired due to transmission losses, acquires a solid meaning and makes possible to model empirically, as proved in chapters 5 and 7.

Thus, the work of this thesis was re-oriented into no-reference transmission-purpose packet-layer and hybrid quality models. Then we faced a universe of possibilities for video encoding settings and transmission schemes: the encoding settings and variables are far from being restricted to the adopted codec, GOP structure, slices and QPs. So, we had to choose what we believe to be a typical configuration setting for encoding. In an industrial setting, the models derived in this thesis must be tuned to the encoding settings adopted by the content provider, and packetizing schemes adopted by the service and network providers, performing new training simulations. Yet, this is not a serious problem, as for instance, the number of encoding configurations (GOP size and structure, bit-rate, and so on) is usually quite limited and, in many cases, only a very small set of parameters can be changed, a limitation imposed by codec manufacturers. Nevertheless, if specific models are needed, they are easy to obtain with the use of neural networks, or even other empirical models. The use of such empirical models is highly convenient in the sense that input variables, topologies, and weights can be adjusted to a wide variety of encoding and transmission conditions, and can be updated in real-time, even using the same types of inputs and targets.

Hence, the quality modeling developed in this thesis becomes a *proof of concept*, which acquires a major importance as it supports automated applications for solving transmission problems. Several applications have been described in this thesis, but the most important one is the ability for the video transmission infrastructure to automatically adapt, according to the traffic and environment conditions, and the predicted QoE at the user-end. The video quality monitor identifies a QoE degradation problem and, using these empirical low-complexity models, it classifies the severity of such problem. Then, it provides this valuable information to other tools (out of the specific scope of this thesis) that will try to solve the problem. Actually, this is the task of engineering.

Annex A: H.264/AVC syntax overview

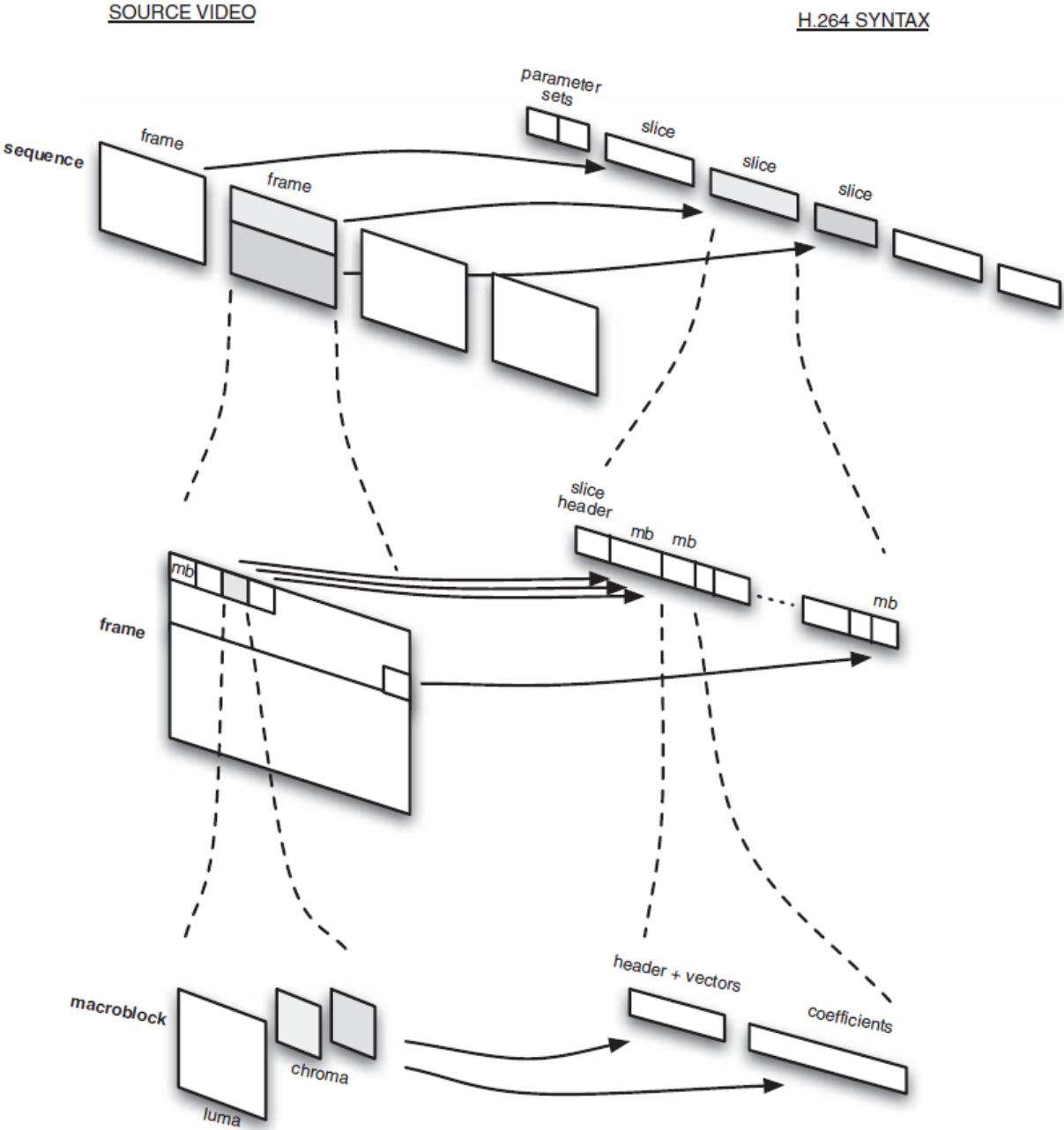


Figure A.1 – H.264/AVC syntax overview (Figure 4.19 from [57]).

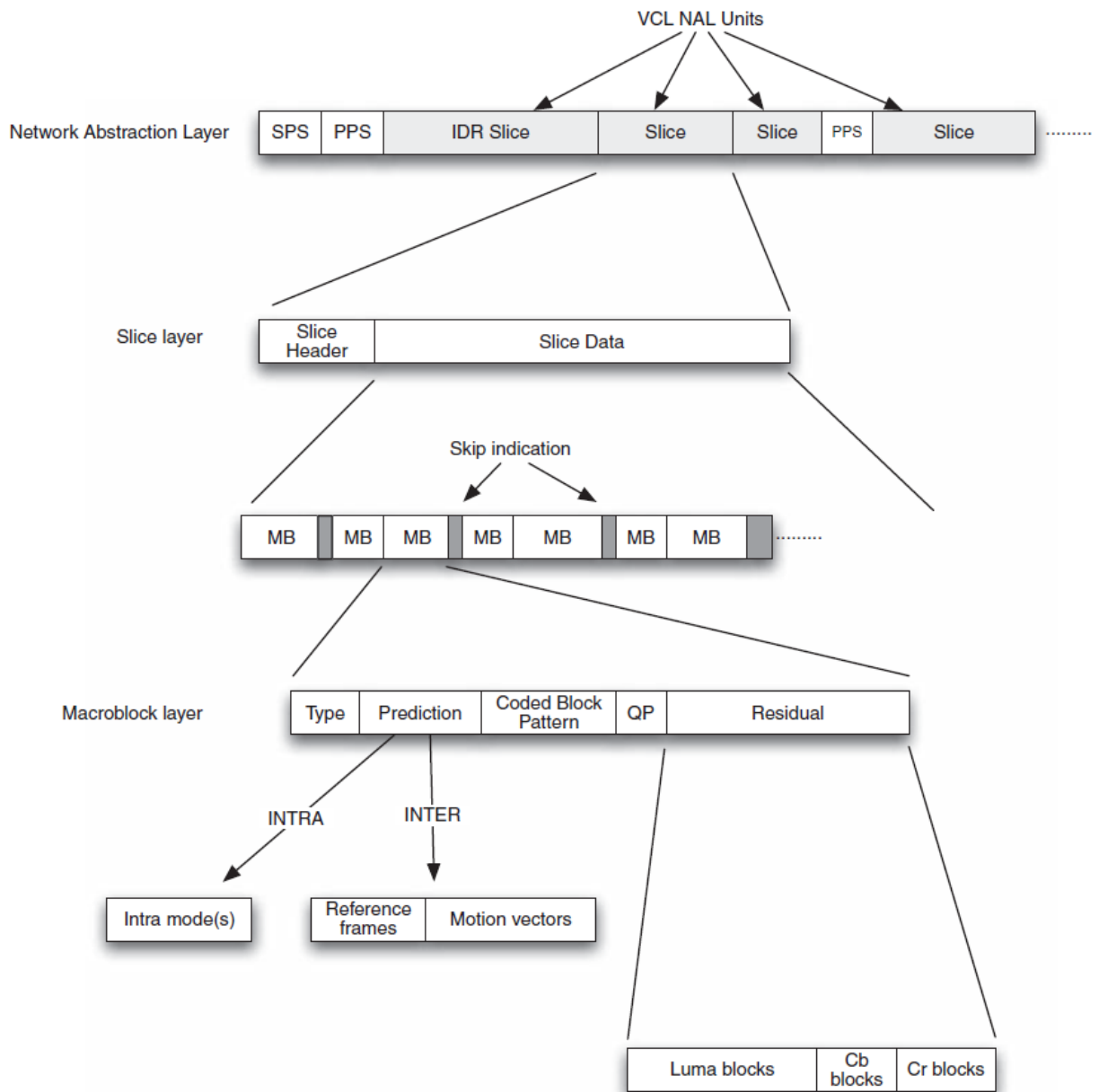


Figure A.2 – H.264/AVC syntax overview (Figure 5.1 from [57]).

Annex B: Trace-file Generator (Matlab script)

```
% Trace-file generator
% by Chamitha de Alwis, adapted by João Soares

%If p is the probability of transferring from Good State to the bad state
%and if r is the probability of transferring from the bad state to the Good
%state, given the p and r values, this code will generate a packet loss
%pattern (with burst losses) and save it to a file named Loss_Pattern.txt.

% 10/04/13 -> maximum burst length defined in maxBL, very important to
% prevent the JM decoder fault.

% p = P(X=1/X=0)
% r = 1 - q = 1 - P(X=1/X=1) = P(X=0/X=1)

%clear all
%clc

% Burst Length (BL) & Packet Loss Rate (PLR) parameter definition
BL = [3 5];
%PLR = [0.001 0.002 0.004 0.007 0.01 0.03 0.05 0.1];
PLR = [0.001 0.002 0.004 0.008 0.02 0.04 0.08 0.2];

maxBL = 7; % maximum burst length

for g = 1:length(BL)
    for h = 1:length(PLR)

        p = 1/(BL(g)*(1/PLR(h) - 1));
        r = 1/BL(g);
        total_packs = 10000;

        check = 100; % check the consistency of the trace-file

        while check >= 10

            loss = 0;
            packets = zeros(1, total_packs);

            for i=1:total_packs
                if loss == 0
                    burst = 0;
                    packets(i) = loss;
                    loss = (rand(1) < p); % P(X=1/X=0), if 1, moves to bad state
                elseif loss == 1
                    burst = burst+1;
                    if burst <= maxBL
                        packets(i) = loss;
                        loss = (rand(1) < (1-r)); % P(X=1/X=1)
                    else
                        packets(i) = 0;
                        loss = 0; % forces to get back to the good state is maxBL is reached
                    end
                end
                else
                    fprintf('error\n');
                    break;
                end
            end

            received_packs = total_packs - nnz(packets);
            theo_pack_loss_rate = 1 - r / (p+r);
            act_pack_loss_rate = 1 - received_packs/total_packs;

            % check the real PLR of the trace-file
            check = abs(theo_pack_loss_rate - act_pack_loss_rate) / theo_pack_loss_rate * 100;

        end

        fid = fopen(['C:\VIDEODATABASE\TOOLS\transmitter_simulator\trace_files\maxBL_7\b' num2str(BL(g)) 'plr'
            num2str(PLR(h)*100)], 'w');
        fprintf(fid, '%d', packets);
        fclose(fid);

        %packets;
        %theo_pack_loss_rate = p / (p+r);
        %act_pack_loss_rate = 1 - received_packs/total_packs;

    end
end
```

Annex C: Camera calibration parameters

	Original view							Virtual view						
	Intrinsic Parameters [A]			Extrinsic Parameters [R t]				Intrinsic Parameters [A]			Extrinsic Parameters [R t]			
Balloons	2241.256	0.0	701.5	1.0	0.0	0.0	0.0	2241.256	0.0	701.5	1.0	0.0	0.0	10.0
	0.0	2241.256	504.5	0.0	1.0	0.0	0.0	0.0	2241.256	504.5	0.0	1.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
Kendo	2241.256	0.0	701.5	1.0	0.0	0.0	0.0	2241.256	0.0	701.5	1.0	0.0	0.0	10.0
	0.0	2241.256	504.5	0.0	1.0	0.0	0.0	0.0	2241.256	504.5	0.0	1.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
Champagne Tower	2969.0	0.0	-832.1011	1.0	0.0	0.0	-1975.0	2969.0	0.0	-795.959	1.0	0.0	0.0	-1875.0
	0.0	2969.0	457.7121	0.0	1.0	0.0	0.0	0.0	2969.0	457.7121	0.0	1.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
Poznan CarPark	1732.8757	0.0	943.2311	1.0	0.0	0.0	0.0	1732.8757	0.0	943.2311	1.0	0.0	0.0	1.593023
	0.0	1739.9089	548.8450	0.0	1.0	0.0	0.0	0.0	1739.9089	548.8450	0.0	1.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0
Newspaper	2929.4940	0.0	307.6333	1.0	0.0	0.0	201.6171	2929.4940	0.0	307.6333	1.0	0.0	0.0	155.2750
	0.0	2922.7064	555.01096	0.0	-1.0	0.0	0.0	0.0	2922.7064	555.01096	0.0	-1.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	1.0	0.0	0.0	-1.0	0.0
Lovebird	2017.8075	0.0	555.4121	1.0	0.0	0.0	193.2942	2017.8075	0.0	555.4121	1.0	0.0	0.0	154.6353
	0.0	2009.3331	385.285	0.0	-1.0	0.0	0.0	0.0	2009.3331	385.285	0.0	-1.0	0.0	0.0
	0.0	0.0	1.0	0.0	0.0	-1.0	0.0	0.0	0.0	1.0	0.0	0.0	-1.0	0.0

Table A.1 – Camera calibration parameters of the videos used in simulations.

	Balloons	Kendo	Champagne Tower	Poznan CarPark	Newspaper	Lovebird
z_{\min}	448.251214	448.251214	2281.357719	-50.191107	3393.977060	1418.292789
z_{\max}	11206.280350	11206.280350	7045.261474	-2760.510889	7542.171244	156012.206895

Table A.2 – Nearest and farthest depth values of depth maps used in simulations.

Annex D: Content description of 3D video used


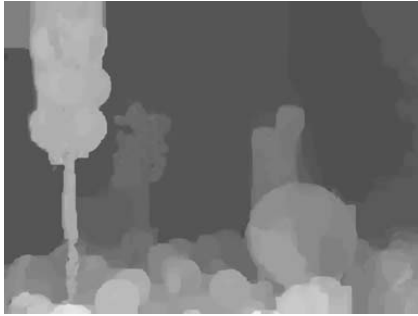
Balloons (cam. 1)	Texture (frame 203)		Depth (frame 203)	
				
	Resolution	Number of frames	Frame Rate	Content description: A man entering and jumping in a big balloon; moving camera, complex motion and moderate depth.
1024x768	300	30 fps		

Table A.3 – Content description of Balloons



Kendo (cam. 1)	Texture (frame 165)		Depth (frame 165)	
				
	Resolution	Number of frames	Frame Rate	Content description: Two men practicing kendo with spectators on the back; moving camera, high motion and moderate depth.
1024x768	300	30 fps		

Table A.4 – Content description of Kendo



Champagne Tower (cam. 39)	Texture (frame 1)		Depth (frame 1)	
				
	Resolution	Number of frames	Frame Rate	Content description: A woman stands next to a cup pyramid and grabs a glass; static camera, moderate motion and high depth.
1280x960	300	30 fps		

Table A.5 – Content description of Champagne Tower



Newspaper (cam. 2)	Texture (frame 245)		Depth (frame 245)	
				
	Resolution	Number of frames	Frame Rate	Content description: Three friends talking around a small table; static camera, moderate motion and moderate depth.
1024x768	300	30 fps		

Table A.6 – Content description of Newspaper

Lovebird1 (cam. 6)	Texture (frame 207)		Depth (frame 207)	
				
	Resolution	Number of frames	Frame Rate	Content description: A boy and a girl walking on a garden, coming closer; static camera, slow motion and high depth.
1024x768	240	30 fps		

Table A.7 – Content description of Lovebird1



Poznan CarPark (cam. 3)	Texture (frame 147)		Depth (frame 147)	
				
	Resolution	Number of frames	Frame Rate	Content description: Men walking in a parking lot and a leaving car; static camera, complex motion and high depth.
1920x1088	250	25 fps		

Table A.8 – Content description of Poznan CarPark

The sequences Balloons, Champagne Tower and Kendo were obtained under permission of Tanimoto Lab at Nagoya University – Japan, available at <http://www.tanimoto.nuee.nagoya-u.ac.jp/> (password protected). The sequence PoznanCarPark was obtained under permission of Poznań University of Technology – Poland, available at <ftp://multimedia.edu.pl/> (password protected). The sequences Newspaper and Lovebird1 were obtained under permission of Gwangju Institute of Science and Technology (GIST) – Republic of Korea, but currently they are not available online.

Annex E: Packet-layer parameter extractor (C++)

The next portion of C++ code, extracted from the developed PL3DVQA software, is the iterative script used to extract the input parameters of the packet-layer quality models of chapter 5, scheme *DOL*, depicted in Table 5.2.

```
class PLI{
private:

    int numSlices, GOPsize, numBframes, numPframes;
    int *slice_size, *slice_num, *slice_lost, *slice_type;
    int **slices_registo;
    string ts_output_file;
    int width, height;

    int num_slices_OK[3], num_slices_KO[3], cumsize_slices_OK[3], cumsize_slices_KO[3], AF, burst_count;
    bool flags[4]; // [loss_P | loss_B | loss_I | event]

    int LB[3], TB[3];
    float PLR[3], AFR, MBL;

    double PSNR, SSIM, MSE;

    double NN_PSNR(); // neural network evaluation functions
    double NN_SSIM();
    double NN_MSE();

public:

    PLI(Parameters *p, GOPstructure *gop); //Constructor:
    ~PLI();// Destructor
    void Extract();
};

void PLI::Extract(){

    ifstream read(ts_output_file); // opens packet-layer output from Naccari's transmitter-simulator
    ofstream write("output.txt");
    int aux1, aux2;

    while(true){ // GOP by GOP

        for (int i = 0; i < 3; i++){ // initialize to zero in the beginning of the GOP
            num_slices_OK[i] = 0;
            num_slices_KO[i] = 0;
            cumsize_slices_OK[i] = 0;
            cumsize_slices_KO[i] = 0;
            flags[i] = 0;
        }

        AF = 0; // number of affected frames
        burst_count = 0;
        flags[3] = 0; // flag_loss_event

        for(int f = 0 ; f < GOPsize ; f++){ // frame-by-frame

            flags[1] = 0; // initialize frame-B flag to zero

            for (int s = 0 ; s < numSlices ; s++){ // slice-by-slice

                // read a line and store the values
                read >> slice_size[s] >> slice_num[s] >> slice_lost[s] >> slice_type[s];

                if(read.eof())
                    goto fim_do_ficheiro; // end-of-file hard stopping condition

                if(slice_lost[s] == 0){
                    // in the case of successful slice reception, register its size in the table
                    slices_registo[slice_type[s]][s] = slice_size[s];
                }
            } // end of slice

            // in this moment, in the first iteration there may be some slices without size,
            // thus they must be estimated from the neighbourhood received slices

            for (int s = 0 ; s < numSlices ; s++){ // slice-by-slice again

                if(slice_lost[s] == 1){ // if it's lost
```

```

if(slices_registo[slice_type[s]][s] > 0){
// if exists the register of size (most probable case)
slice_size[s] = slices_registo[slice_type[s]][s];
}
}else{ // non-zero-mean of the values of the table
aux1 = 0;
aux2 = 0;

for(int i = 0; i < numSlices ; i++){
if(slices_registo[slice_type[s]][i] > 0){
aux1 += slices_registo[slice_type[s]][i];
aux2++;
}
}

slices_registo[slice_type[s]][s] = aux1/aux2;
slice_size[s] = slices_registo[slice_type[s]][s];
}

cumsize_slices_KO[slice_type[s]] += slice_size[s];
num_slices_KO[slice_type[s]] ++;

flags[slice_type[s]] = 1;
flags[3] = 1;

} else { // if the slice has been received

cumsize_slices_OK[slice_type[s]] += slice_size[s];
num_slices_OK[slice_type[s]] ++;

if(flags[3] == 1){
burst_count++; // increments burst counter
flags[3] = 0;
}

}

} // end of slice

// increment AF, if the frame is affected (any slice)

if(flags[0] || flags[1] || flags[2]) // if there's any active flag_loss
AF++;

} // end of frame

// Computation of PLR, LB e TB for each type of frame
PLR[0] = (float) num_slices_KO[0] / (numPframes*numSlices);
PLR[1] = (float) num_slices_KO[1] / (numBframes*numSlices);
PLR[2] = (float) num_slices_KO[2] / numSlices;

for(int i = 0; i < 3; i++){
LB[i] = cumsize_slices_KO[i];
TB[i] = cumsize_slices_KO[i] + cumsize_slices_OK[i];
}

if(burst_count > 0)
MBL = (float) (num_slices_KO[0] + num_slices_KO[1] + num_slices_KO[2]) / burst_count;
else
MBL = 0;

AFR = (float) AF/GOPsize;

write << PLR[0] << '\t' << PLR[1] << '\t' << PLR[2] << '\t' << LB[0] << '\t' << LB[1] << '\t' << LB[2] <<
'\t' << TB[0] << '\t' << TB[1] << '\t' << TB[2] << '\t' << AFR << '\t' << MBL << '\t';

// calls neural network evaluation functions
PSNR = NN_PSNR();
SSIM = NN_SSIM();
MSE = NN_MSE();

write << PSNR << '\t' << SSIM << '\t' << MSE << endl; // end of output line

} // end of GOP

fim_do_ficheiro;;

read.close();
write.close();

}

```

Annex F: GOP structure detector (C++)

The next portion of C++ code is the script used for GOP structure detection, extracted from the developed PL3DVQA simulation software.

```
class GOPstructure{
private:
    int numSlices, GOPsize;
    int numFrames[2];
    string ts_output_file;
public:
    //Constructor:
    GOPstructure(Parameters *p);

    // Destructor
    ~GOPstructure();

    void detect_GOP();

    int get_numSlices(){return numSlices;}
    int get_GOPsize(){return GOPsize;}
    int get_numBframes(){return numFrames[1];}
    int get_numPframes(){return numFrames[0];}
};

void GOPstructure::detect_GOP(){

    ifstream read;
    read.open(ts_output_file); // opens the Naccari's transmitter-simulator output file

    ofstream write;
    write.open("GOPstructure.txt");

    int aux[4]; // auxilliary variables used to store values as the file is being read
    while(1){

        read >> aux[0] >> aux[1] >> aux[2] >> aux[3];

        if(aux[1] > 0 && aux[3] == 2)
            break;

        if(aux[1] == 0) // if is the first frame, increments the number of slices
            numSlices++;
        else if(aux[3] < 2)
            numFrames[aux[3]]++;

        GOPsize++; // increments GOP's number of packets
    }

    GOPsize /= numSlices; // number of GOP frames
    numFrames[0] /= numSlices; // number of P-frames
    numFrames[1] /= numSlices; // number of B-frames

    write << GOPsize << '\t' << numFrames[0] << '\t' << numFrames[1] << '\t' << numSlices << endl;

    read.close();
    write.close();
}
}
```

Annex G: Packet-Layer specifications and overall quality of the 3D videos used in the subjective tests

Rank (w)	Packet-Loss-Rate (PLR)				Mean Burst Length (MBL)	Affected Frame Rate (AFR)	Average PSNR	Average SSIM
	Overall	P-frames	B-frames	I-frames				
1	0,42%	0,29%	0,13%	0,00%	5,0	2,00%	51	0,9998
2	0,92%	0,42%	0,50%	0,00%	3,6	14,30%	46	0,9991
3	3,96%	0,75%	3,00%	0,21%	4,5	33,00%	36	0,9902
4	8,38%	2,08%	6,04%	0,25%	3,4	68,00%	34	0,9861
5	18,50%	4,63%	13,46%	0,42%	4,0	91,30%	30	0,9723

Table A.9 – Specifications and overall quality of the Balloons sample set

Rank (w)	Packet-Loss-Rate (PLR)				Mean Burst Length (MBL)	Affected Frame Rate (AFR)	Average PSNR	Average SSIM
	Overall	P-frames	B-frames	I-frames				
1	0,46%	0,08%	0,38%	0,00%	2,7	7,00%	56	0,9999
2	1,04%	0,54%	0,50%	0,00%	5,0	24,30%	45	0,9996
3	4,38%	0,96%	3,33%	0,08%	2,5	75,00%	34	0,9945
4	8,46%	1,92%	6,29%	0,25%	3,3	61,60%	30	0,9891
5	19,38%	5,33%	13,63%	0,42%	4,2	90,00%	25	0,9753

Table A.10 – Specifications and overall quality of the Champagne Tower sample set

Rank (w)	Packet-Loss-Rate (PLR)				Mean Burst Length (MBL)	Affected Frame Rate (AFR)	Average PSNR	Average SSIM
	Overall	P-frames	B-frames	I-frames				
1	0,54%	0,00%	0,54%	0,00%	4,3	1,30%	50	0,9998
2	1,04%	0,33%	0,71%	0,00%	5,0	20,60%	42	0,9967
3	4,04%	0,88%	3,17%	0,00%	2,4	58,30%	36	0,9938
4	8,04%	2,21%	5,63%	0,21%	3,2	55,00%	33	0,9886
5	19,54%	4,63%	14,00%	0,92%	4,3	90,66%	26	0,9547

Table A.11 – Specifications and overall quality of the Kendo sample set

Rank (w)	Packet-Loss-Rate (PLR)				Mean Burst Length (MBL)	Affected Frame Rate (AFR)	Average PSNR	Average SSIM
	Overall	P-frames	B-frames	I-frames				
1	0,50%	0,05%	0,45%	0,00%	5,0	9,60%	53	0,9996
2	1,20%	0,25%	0,95%	0,00%	3,0	23,20%	46	0,9985
3	5,00%	1,65%	3,20%	0,15%	5,0	57,20%	37	0,9797
4	8,10%	1,90%	5,90%	0,30%	2,9	69,20%	34	0,9782
5	19,00%	5,50%	12,90%	0,60%	3,0	93,60%	30	0,9413

Table A.12 – Specifications and overall quality of the Poznan CarPark sample set

Annex H: Discontinuity measure at macroblock edges (Matlab script)

```
persistent linhas;
persistent colunas;
B = 16; % macroblock side pixel-length

if isempty(linhas)
    linhas = logical(repmat([1, zeros(1, B-1)]', dims(1)/B, dims(2)));
    linhas(1, :) = 0;
end
if isempty(colunas)
    colunas = logical(repmat([1, zeros(1, B-1)], dims(1), dims(2)/B));
    colunas(:, 1) = 0;
end

TH1 = 5; % threshold for discontinuity detection at block edges

block_effect = zeros(numSlices,numfrm);

for k=1:numfrm % frame-by-frame

    if sum(afectacao(:,k)) > 0 % if is there any affected slice in this frame

        % frame replication with a single-line displacement to the bottom, and a single column to the right
        Yd_shiftdown = [zeros(1,dims(2)) ; Y2{k}(1:end-1, :)];
        Yd_shiftright = [zeros(1,dims(1))' ,Y2{k}(:, 1:end-1)];

        % pixel-wise calculation of the differences between the original and displaced frames,
        % and multiplication by the grid mask
        aux_linhas_d = (double(Y2{k}) - double(Yd_shiftdown)).*linhas;
        aux_colunas_d = (double(Y2{k}) - double(Yd_shiftright)).*colunas;

        % pixel-wise difference energy measure (only if the threshold TH1 is exceeded)
        mbedge_d = (aux_linhas_d.^2) .* (abs(aux_linhas_d) > TH1) + ...
            (aux_colunas_d.^2) .* (abs(aux_colunas_d) > TH1);

        for s=1:numSlices
            if(afectacao(s,k) == 1)

                % overall slice discontinuity measure: average and square-root
                block_effect(s,k) = sqrt( sum(sum(mbedge_d(dims(1)/numSlices*(s-1)+1 : dims(1)/numSlices*(s) ,
:))) / sum(sum(logical(linhas+colunas))) * numSlices );

            else
                block_effect(s,k) = 0;
            end
        end
    else
        block_effect(:,k) = 0;
    end
end

end
```

Annex I: Examples of the media-layer quality monitor performance for scheme *DOL*

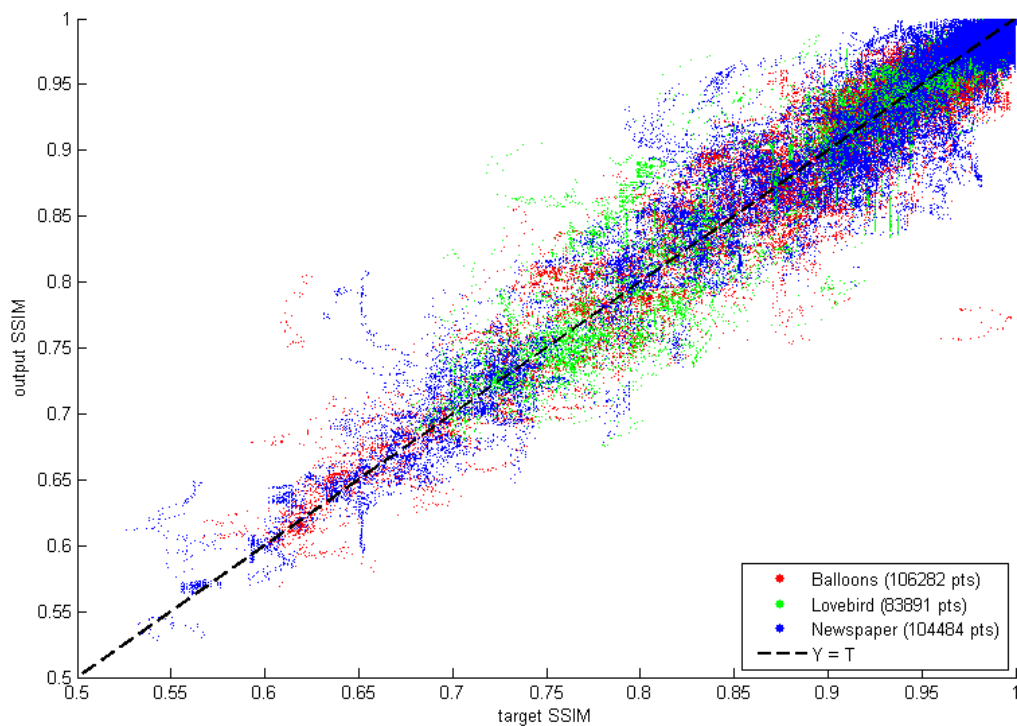


Figure A.3 – Performance of the media-layer 3D-VQM (Architecture 1) for scheme *DOL*, using the actual predicted SSIM from the previous co-located slice as the feedback input. PLCC = 0.97079

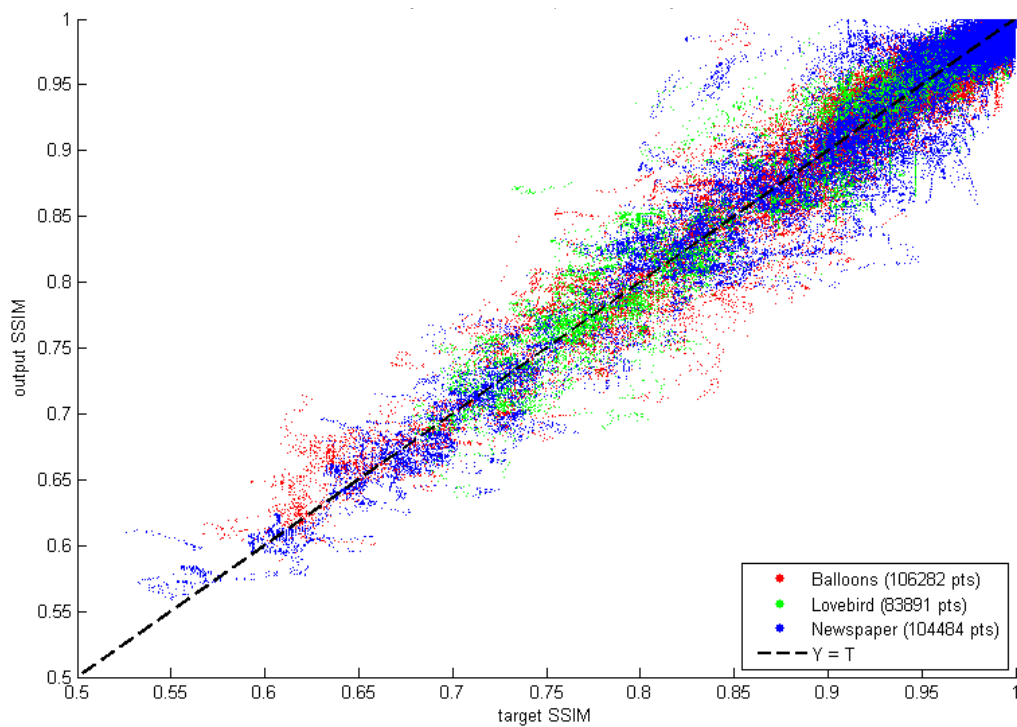


Figure A.4 – Performance of the media-layer 3D-VQM (Architecture 2) for scheme *DOL*, using the actual predicted SSIM from the previous co-located slice as the feedback input. PLCC = 0.97778

Bibliography

- [1] Cisco Visual Networking Index, “The Zettabyte Era — Trends and Analysis,” 2013.
- [2] A. R. Reibman and John G. Apostolopoulos, “The Challenge of Estimating Video Quality in Video Communication Applications,” *Signal Processing Magazine, IEEE*, vol. 29, no. 2, pp. 160–158, 2012.
- [3] M. G. Martini, M. Mazzotti, and C. Lamy-bergot, “Content Adaptive Network Aware Joint Optimization of Wireless Video Transmission,” *Communications Magazine, IEEE*, vol. 45, no. 1, pp. 84–90, 2007.
- [4] Z. Song, H. Wang, Y. Wen, D. Wu, and K. Lee, “Depth-color based 3D image transmission over wireless networks with QoE provisions,” *Computer Communications*, vol. 35, no. 15, pp. 1838–1845, Sep. 2012.
- [5] P. Pérez and N. García, “Lightweight Multimedia Packet Prioritization Model for Unequal Error Protection,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 132–138, 2011.
- [6] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman, “A versatile model for packet loss visibility and its application to packet prioritization,” *IEEE transactions on image processing*, vol. 19, no. 3, pp. 722–735, Mar. 2010.
- [7] C. T. E. R. Hewage, S. Nasir, S. T. Worrall, and M. G. Martini, “Prioritized 3D Video Distribution over IEEE 802.11e,” in *Future Network & Mobile Summit 2010 Conference Proceedings*, 2010, pp. 1–9.
- [8] S. Cardeal, F. Neves, S. Soares, F. Tavares, and P. Assuncao, “ArQoS®: System to monitor QoS/QoE in VoIP,” *2011 IEEE EUROCON - International Conference on Computer as a Tool*, pp. 1–2, Apr. 2011.
- [9] A. K. Moorthy and A. C. Bovik, “Visual quality assessment algorithms: what does the future hold?,” *Multimedia Tools and Applications*, vol. 51, no. 2, pp. 675–696, Oct. 2010.
- [10] ITU, “Recommendation ITU-R BT.500-11: Methodology for the subjective assessment of the quality of television pictures.” pp. 1–48, 2002.
- [11] ITU, “Recommendation ITU-R BT.2021: Subjective methods for the assessment of stereoscopic 3DTV systems.” 2012.
- [12] Q. Huynh-Thu, P. Le Callet, and M. Barkowsky, “Video Quality Assessment From 2D To 3D – Challenges And Future Trends,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, 2010, pp. 4025 – 4028.
- [13] B. Feitor, P. Assunção, L. Cruz, J. Soares, and R. Marinheiro, “Objective quality prediction model for lost frames in 3D video over TS,” in *IEEE ICC 2013 Conference Proceedings*, 2013.

- [14] B. Feitor, P. Assunção, L. Cruz, J. Soares, and R. Marinheiro, “No-Reference Quality Models for Single Frame Loss in 3D Video,” in *Proceedings of the 9th Conference on Telecommunications, Castelo-Branco, Portugal*, 2013, pp. 73–76.
- [15] D. Minoli, *3DTV CONTENT CAPTURE , ENCODING AND TRANSMISSION: Bulding the transport infraestrutura for commercial services*. 2010.
- [16] C. Fehn, “A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR),” in *Visualization, Imaging, and Image Processing*, 2003, vol. 396, pp. 84–89.
- [17] A. Tikanm, A. Gotchev, A. Smolic, and K. Müller, “Quality assessment of 3D video in rate allocation experiments,” in *Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on*, 2008, pp. 1–4.
- [18] Y.-S. Kang and Y.-S. Ho, “Disparity map generation for color image using TOF depth camera,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2011, vol. 1, no. c, pp. 1–4.
- [19] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, “Temporal filtering for depth maps generated by Kinect depth camera,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2011, pp. 1–4.
- [20] J.-I. Park and S. Inoue, “Acquisition of Sharp Depth Map from Multiple Cameras,” *Signal Processing: Image Communication*, no. Special Issue on 3D Video technology, pp. 1–18, 1997.
- [21] C. Liu and L. Christopher, “Depth map estimation from motion for 2D to 3D conversion,” *2012 IEEE International Conference on Electro/Information Technology*, vol. 1, pp. 1–4, May 2012.
- [22] L. He-jian, T. Guo-wei, Z. Zhao-yang, A. Ping, M. Ran, W. Jian-wei, and W. Fu-qiong, “Hardware solution of real-time depth estimation based on stereo vision,” *2012 International Conference on Audio, Language and Image Processing*, pp. 39–44, Jul. 2012.
- [23] K. Müller, P. Merkle, and T. Wiegand, “3-D Video Representation Using Depth Maps,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, 2011.
- [24] J. Greengrass, J. Evans, and A. C. Begen, “Not All Packets Are Equal, Part I: Streaming Video Coding and SLA Requirements,” *Internet Computing, IEEE*, vol. 13, no. 1, pp. 70–75, 2009.
- [25] Q. Zhang, C. H. Cui, K. N. Ngan, and Y. Liu, “Depth estimation and view synthesis for narrow-baseline video,” *2012 IEEE International Symposium on Circuits and Systems*, pp. 1883–1886, May 2012.
- [26] M. Köppel, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegand, “Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering,” in *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, 2010, pp. 1809–1812.

- [27] M. Gotfryd, K. Wegner, and M. Domański, “View synthesis software and assessment of its performance,” 2008.
- [28] G. Shen, W.-S. Kim, S. K. Narang, A. Ortega, J. Lee, and H. Wey, “Edge-adaptive transforms for efficient depth map coding,” in *Picture Coding Symposium (PCS), 2010*, 2010, pp. 566–569.
- [29] D. V. S. X. De Silva, W. A. C. Fernando, and H. K. Arachchi, “A new mode selection technique for coding Depth maps of 3D video,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 686–689.
- [30] H. Urey, K. V Chellappan, E. Erden, and P. Surman, “State of the Art in Stereoscopic and Autostereoscopic Displays,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 540–555, Apr. 2011.
- [31] A. Takahashi, D. Hands, and V. Barriac, “Standardization Activities in the ITU for a QoE Assessment of IPTV,” *Communications Magazine, IEEE*, vol. 46, no. 2, pp. 78–84, 2008.
- [32] ITU, “Recommendation ITU-T J.143: User requirements for objective perceptual video quality measurements in digital cable television.” pp. 1–15, 2000.
- [33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity.,” *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 13, no. 4, pp. 600–12, Apr. 2004.
- [34] M. Vranješ, S. Rimac-Drlje, and K. Grgić, “Review of objective video quality metrics and performance comparison using different databases,” *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 1–19, Jan. 2013.
- [35] K. Zeng and Z. Wang, “3D-SSIM for video quality assessment,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 621–624.
- [36] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, “Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison,” *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 165–182, 2011.
- [37] M.-J. Chen, D.-K. Kwon, and A. C. Bovik, “Study of subject agreement on stereoscopic video quality,” *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 173–176, Apr. 2012.
- [38] C. T. E. R. Hewage and M. G. Martini, “Quality of Experience for 3D Video Streaming,” *Communications Magazine, IEEE*, vol. 51, no. 5, pp. 101–107, 2013.
- [39] M. H. Pinson and S. Wolf, “A New Standardized Method for Objectively Measuring Video Quality,” *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, Sep. 2004.
- [40] K. Seshadrinathan and A. C. Bovik, “Motion tuned spatio-temporal quality assessment of natural videos.,” *IEEE transactions on image processinga publication of the IEEE Signal Processing Society*, vol. 19, no. 2, pp. 335–50, Feb. 2010.

- [41] J. You, J. Korhonen, and A. Perkis, "Balancing Attended and Global Stimuli in Perceived Video Quality Assessment," *Multimedia, IEEE Transactions on*, vol. 13, no. 6, pp. 1269–1285, 2011.
- [42] VQEG, "Final Report From The Video Quality Experts Group On The Validation Of Objective Models Of Video Quality Assessment, Phase II," 2003.
- [43] G. Valenzise, S. Magni, M. Tagliasacchi, and S. Tubaro, "No-Reference Pixel Video Quality Monitoring of Channel-Induced Distortion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 605–618, 2012.
- [44] C. T. E. R. Hewage, S. T. Worrall, S. Dogan, S. Villette, and A. M. Kondoz, "Quality Evaluation of Color Plus Depth Map-Based Stereoscopic Video," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 2, pp. 304–318, 2009.
- [45] M. Solh, G. Alregib, and J. M. Bauza, "3VQM: A vision-based quality measure for DIBR-based 3D videos," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, 2011, pp. 1–6.
- [46] P. Joveluro, H. Malekmohamadi, W. A. C. Fernando, and A. M. Kondoz, "Perceptual Video Quality Metric for 3D video quality assessment," *2010 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 1–4, Jun. 2010.
- [47] C. Sun, X. Liu, and W. Yang, "An Efficient Quality Metric for DIBR-based 3D Video," *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, pp. 1391–1394, Jun. 2012.
- [48] S. L. P. Yasakethu, S. T. Worrall, D. V. S. X. De Silva, W. a. C. Fernando, and A. M. Kondoz, "A compound depth and image quality metric for measuring the effects of packet loss on 3D video," *2011 17th International Conference on Digital Signal Processing (DSP)*, pp. 1–7, Jul. 2011.
- [49] A. Maalouf, M. Larabi, B. Marie, P. O. Box, and F. Chasseneuil, "CYCLOPA Stereo Color Image Quality Assessment Metric," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 1161–1164.
- [50] C. T. E. R. Hewage, M. G. Martini, and S. Member, "Edge-Based Reduced-Reference Quality Metric for 3-D Video Compression and Transmission," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 5, pp. 471–482, 2012.
- [51] G. Nur and G. B. Akar, "An abstraction based reduced reference depth perception metric for 3D video," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 625–628.
- [52] P. Sazzad, S. Yamanaka, Y. Kawayoke, and Y. Horita, "Stereoscopic image quality prediction," in *Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on*, 2009, pp. 180–185.
- [53] M. Solh and G. Alregib, "A no-reference quality measure for DIBR-based 3D videos," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2011, pp. 1–6.

- [54] E. Bosc, P. Le Callet, L. Morin, and M. Pressigout, “An edge-based structural distortion indicator for the quality assessment of 3D synthesized views,” in *Picture Coding Symposium (PCS)*, 2012, pp. 249–252.
- [55] A. Mittal, A. K. Moorthy, J. Ghosh, and A. C. Bovik, “Algorithmic assessment of 3D quality of experience for images and videos,” in *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, 2011, pp. 338–343.
- [56] “ITU-T H.264: Advanced video coding for generic audiovisual services.” pp. 1–680, 2012.
- [57] I. E. Richardson, *The H.264 advanced video compression standard*. 2010.
- [58] R. Even and Y. Wang, “RTP Payload Format for H.264 Video,” 2011.
- [59] “ITU-T T.81 Information Technology – Digital Compression And Coding Of Continuous-Tone Still Images – Requirements And Guidelines.” 1993.
- [60] H. Sanneck and G. Carle, “A Framework Model for Packet Loss Metrics Based on Loss Runlengths,” in *SPIE/ACM SIGMM Multimedia Computing and Networking Conference, Proceedings of the*, 2000, vol. 2000, no. January 2000, pp. 1–11.
- [61] G. Haßlinger and O. Hohlfeld, “The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet,” *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference*, 2008.
- [62] R. Skupin, C. Hellge, T. Schierl, and T. Wiegand, “Packet level video quality evaluation of extensive H.264/AVC and SVC transmission simulation,” *Journal of Internet Services and Applications*, vol. 2, no. 2, pp. 129–138, Jun. 2011.
- [63] M. Naccari, “H.264/AVC bitstream transmission simulator,” no. December. 2008.
- [64] L. Trudeau, S. Coulombe, and S. Pigeon, “Pixel domain referenceless visual degradation detection and error concealment for mobile video,” in *18th IEEE International Conference on Image Processing*, 2011, pp. 2229–2232.
- [65] H. Hui and C. Tie-yong, “An Efficient Error Concealment Algorithm for Intra-frames of H.264,” in *Communication Technology (ICCT), 2010 12th IEEE International Conference on*, 2010, pp. 576–579.
- [66] C. T. E. R. Hewage, S. T. Worrall, S. Dogan, and A. M. Kondo, “A Novel Frame Concealment Method for Depth Maps Using Corresponding Colour Motion Vectors,” *2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 149–152, May 2008.
- [67] B. Yan and J. Zhou, “Efficient Frame Concealment for Depth Image-Based 3-D Video Transmission,” *Multimedia, IEEE Transactions on*, vol. 14, no. 3, pp. 936–945, 2012.
- [68] Y. Zhang, X. Xiang, D. Zhao, S. Ma, and W. Gao, “Packet Video Error Concealment With Auto Regressive Model,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 12–27, Jan. 2012.

- [69] J. Mochnác and S. Marchevský, “Error Concealment Scheme Implemented in H.264/AVC,” in *ELMAR, 2008. 50th International Symposium*, 2008, no. September, pp. 13–16.
- [70] H. Shukla, “Design Of An Error Concealment Scheme For H.264/AVC Video Bitstream,” 2012.
- [71] P. Frank and J. Incera, “A Neural Network Based Test Bed for Evaluating the Quality of Video Streams in IP Networks,” *Electronics, Robotics and Automotive Mechanics Conference (CERMA’06)*, vol. 1, no. 1, pp. 178–183, Sep. 2006.
- [72] B. E. Akoa, E. Simeu, and F. Lebowsky, “Using Artificial Neural Network for Automatic Assessment of Video Sequences,” in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, 2013, pp. 285–290.
- [73] J. Choe, K. Lee, C. Lee, and S. Korea, “No-reference video quality measurement using neural networks,” in *Digital Signal Processing, 2009 16th International Conference on*, 2009, pp. 1–4.
- [74] D. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [75] A. Boev, A. Gotchev, and K. Egiazarian, “Crosstalk Measurement Methodology for Auto-Stereoscopic Screens,” in *3DTV Conference, 2007*, no. 4, pp. 1–4.
- [76] Q. Huynh-Thu, M. Barkowsky, and P. Le Callet, “The Importance of Visual Attention in Improving the 3D-TV Viewing Experience: Overview and New Perspectives,” *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 421–431, Jun. 2011.
- [77] Y. Yi, J. Ding, and J. Lai, “A novel video salient object extraction method based on visual attention,” *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 45–54, Jan. 2013.
- [78] J. Wang, M. P. Da Silva, P. Le Callet, and V. Ricordel, “Computational model of stereoscopic 3D visual saliency,” *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, vol. 22, no. 6, pp. 2151–65, Jun. 2013.
- [79] C. Spearman, “The Proof and Measurement of Association between Two Things,” *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.