



Politecnico di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

Video Streaming Systems in Immersive mode = Sistemi di Streaming Video in modalità immersiva

This is a PhD Thesis

Original Citation:

Video Streaming Systems in Immersive mode = Sistemi di Streaming Video in modalità immersiva / Ribezzo, Giuseppe. - ELETTRONICO. - (2021). [10.60576/poliba/iris/ribezzo-giuseppe_phd2021]

Availability:

This version is available at <http://hdl.handle.net/11589/226740> since: 2023-07-11

Published version

Politecnico di Bari
DOI: 10.60576/poliba/iris/ribezzo-giuseppe_phd2021

Terms of use:

Altro tipo di accesso

(Article begins on next page)



DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING
ELECTRICAL AND INFORMATION ENGINEERING
PHD PROGRAM
S.S.D. ING-INF/04

Final Dissertation

Video Streaming Systems in Immersive mode

by

Giuseppe RIBEZZO

Supervisors:

Prof. Ing. Saverio MASCOLO

Prof. Ing. Luca DE CICCO

Coordinator of the Ph.D Program:

Prof. Ing. Luigi Alfredo GRIECO

Course XXXIII, 01/11/2018 - 31/12/2020



Politecnico
di Bari

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

ELECTRICAL AND INFORMATION ENGINEERING

PHD PROGRAM

S.S.D. ING-INF/04

Final Dissertation

Video Streaming Systems in Immersive mode

by

Giuseppe RIBEZZO

Supervisor:

Prof. Ing. Saverio MASCOLO

Prof. Ing. Luca DE CICCO

Coordinator of the Ph.D Program:

Prof. Ing. Luigi Alfredo GRIECO

Course XXXIII, 01/11/2018 - 31/12/2020

Al Magnifico Rettore
del Politecnico di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Il sottoscritto Giuseppe RIBEZZO nato a Francavilla Fontana (BR) il 17/02/1989 residente a Brindisi in via Gioacchino ROSSINI, 31/E, e-mail ribes170289@gmail.com iscritto al 3 anno di Corso di Dottorato di Ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo XXXIII ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

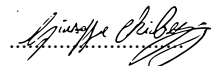
Video Streaming Systems in Immersive mode

DICHIARA

1. di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2. di essere iscritto al Corso di Dottorato di ricerca in INGEGNERIA ELETTRICA E DELL'INFORMAZIONE ciclo XXXIII, corso attivato ai sensi del "Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari", emanato con D.R. n.286 del 01.07.2013;
3. di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4. di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archiverà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito <http://www.creativecommons.it/Licenze>), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5. che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle consegnate/inviolate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6. che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7. che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali ed economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data BARI, 04/11/2020

Firma



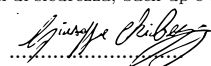
Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data BARI, 04/11/2020

Firma





*Al Magnifico Rettore
del Politecnico di Bari*

RICHIESTA DI EMBARGO


Sottoscrivere solo nel caso in cui si intenda auto-archiviare la tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica POLIBA-IRIS (<https://iris.poliba.it>) non in modalità "Accesso Aperto", per motivi di segretezza e/o di proprietà dei risultati e/o informazioni sensibili o sussistano motivi di segretezza e/o di proprietà dei risultati e informazioni di Enti esterni o Aziende private che hanno partecipato alla realizzazione della ricerca.

Il sottoscritto Giuseppe RIBEZZO nato a Francavilla Fontana (BR) il 17/02/1989 residente a Brindisi in via Gioacchino ROSSINI, 31/E, e-mail ribes170289@gmail.com iscritto al 3 anno di Corso di Dottorato di Ricerca in INGEGNERIA DELL'INFORMAZIONE ciclo XXXIII, Autore della tesi di dottorato dal titolo **Video Streaming Systems in Immersive mode** e ammesso a sostenere l'esame finale:

AUTORIZZA

Il Politecnico di Bari a pubblicare nell'Archivio Istituzionale di Ateneo ad accesso aperto il testo completo della tesi depositata.

Luogo e data BARI, 04/11/2020

Firma Dottorando 

Firma Relatore

To my family, that supported me since from the beginning of my studies in dedicating time and efforts in studying and increasing my knowledge.

Acknowledgements

I would like to acknowledge my tutors, Prof. Ing. Saverio Mascolo and Prof. Ing. Luca De Cicco, for all the efforts spent in assisting me on developing each part of the research topic included in this thesis.

Contents

List of Figures	vii
List of Tables	ix
Introduction: Dissertation Overview	xi
Personal Scientific Contributions	xv
1 The Immersive Streaming technology	1
1.1 Historical background	1
1.2 Today's Immersive applications	5
1.3 Overview on the Streaming Protocols Ecosystem	6
1.3.1 HTTP Live Streaming	9
1.3.2 MPEG Dynamic Adaptive Streaming over HTTP	10
1.3.2.1 The Media Presentation Description (MPD) Manifest	10
1.3.2.2 Media segments format	12
1.3.3 MPEG Common Media Application Format	13
1.3.4 Coding Standards	14
1.3.4.1 Moving Pictures Experts Group (MPEG) High Efficiency Video Coding	15
1.4 Algorithms for the Quality of Experience (QoE) estimation	19
1.4.1 Subjective QoE metrics	19
1.4.2 Objective QoE metrics	21
1.4.2.1 A digression about the Support Vector Machine (SVM) algorithm	25
1.5 Technologies for Immersive Video Streaming	27

CONTENTS

1.5.1	Omnidirectional Video (OV) projection formats	30
1.5.1.1	Viewport independent mappings	30
1.5.1.2	Viewport dependent mappings	37
1.5.2	Tiled Streaming	40
1.5.2.1	MPEG DASH Spatial Relationship Description	42
1.5.2.2	MPEG HEVC Motion Constrained Tile Set	44
1.6	A standard for Virtual Reality: MPEG Omnidirectional Media Format	45
1.6.0.1	OMAF ISOBMFF and DASH extension	47
1.7	Control systems for adaptive video streaming	50
1.7.1	Rate-based approaches	51
1.7.2	Level-based approaches	52
1.7.3	Control problem on Immersive Video Streaming	54
2	Reducing the Network Bandwidth Requirements for 360 Immersive Video Streaming	57
2.1	Background	57
2.2	Proposed Approach	59
2.3	Methodology	61
2.4	Considered scenarios	62
2.5	Results	63
2.6	Final considerations on the proposed codec-agnostic solution for bitrate reduction	66
3	A Dynamic Adaptive Streaming over HTTP (DASH)-compliant immersive streaming architecture	67
3.1	The Proposed Immersive Platform	67
3.1.1	DASH-compliant Server Design	68
3.1.2	Viewport adaptive Client	69
3.1.2.1	Quality Selection Algorithm	69
3.1.2.2	View Selection Algorithm	70
3.2	Experimental Evaluation on a real Use Case	72
3.3	Experimental Results	75
3.4	Final considerations	77

4 Bitrate Reduction for Immersive Streaming: Comparing Variable Quantization Parameter (VQP) and Variable Resolution (VRES) Approaches	79
4.1 Introduction	79
4.2 Related Works	80
4.3 Bitrate Reduction Techniques	81
4.3.1 VQP approach	82
4.3.2 VRES approach	83
4.4 Methodology	83
4.5 Results	85
4.5.1 Bitrate reduction	85
4.5.2 Visual quality as a function of the user’s head position	87
4.5.3 Visual quality as a function of video content	89
4.6 Final considerations about VRES and VQP bitrate reduction schemes	89
5 TAPAS-360: a Tool for the Design and Experimental Evaluation of 360 Video Streaming Systems	93
5.1 Introduction	93
5.2 TAPAS-360	95
5.2.1 Tapas360Player	95
5.2.2 Parser360	96
5.2.3 MediaEngine	96
5.2.4 QualityController	97
5.2.5 ViewController	98
5.2.6 HMDEmulator	99
5.3 Use Cases	100
5.3.1 2D video streaming	100
5.3.2 Viewport-adaptive streaming	100
5.3.3 Subjective and Objective Quality of Experience evaluations	101
5.4 Summary	101
6 Conclusions and Future Research Directions	103
References	107

CONTENTS

List of Figures

1.1	The <i>The Sword of Damocles</i>	2
1.2	VPL research Data Gloves and EyePhone.	4
1.3	The Virtuality continuum.	5
1.4	High-level view of Extended Reality applications.	5
1.5	RTSP/RTP sequence diagram.	7
1.6	PDS sequence diagram.	8
1.7	Diagram of the Adaptive Bit Rate (ABR) streaming.	9
1.8	MPD Data Model.	11
1.9	Diagram summarizing High Efficiency Video Coding (HEVC) encoding.	16
1.10	Diagram of the SSIM measurement system [1].	23
1.11	<i>Stitching</i> process of 360 panorama.	28
1.12	ERP sphere-to-plane mapping	31
1.13	Omnidirectional scene in ERP format [2].	31
1.14	Omnidirectional scene in EAP format [2]. Please note as straight lines are curved.	32
1.15	Pipeline for generating OV contents in CMP format.	33
1.16	JVET CMP layout.	33
1.17	Two-dimensional illustration of the CMP projection process	34
1.18	<i>Rhombic Dodecahedron</i> video projection workflow.	35
1.19	EAC/ACP linearizing function.	36
1.20	Segmentation approach.	36
1.21	The SSP mapping proposed by JVET.	37
1.22	Barrel layout of a 360 video [3].	38
1.23	<i>Pyramid projection</i> video production pipeline [4].	39

LIST OF FIGURES

1.24	<i>Offset Cubemap Projection</i> [5].	40
1.25	Example diagram of the Tiled Streaming approach.	41
1.26	A sketch of the MPD extended by Spatial Relationship Description (SRD).	43
2.1	Approach to generate the i -th Region of Interest (RoI) representation	60
2.2	Average Peak Signal-to-Noise Ratio (PSNR) function of the viewport yaw angle	64
2.3	Average Structural Similarity Index Measurement (SSIM) function of the viewport yaw angle	64
2.4	SSIM as a function of the Bitrate reduction percentage	64
3.1	The viewport adaptive immersive streaming architecture	68
3.2	View selection algorithm in the case of $N = 3$ or $N = 4$ views	71
3.3	The proposed delivery system architecture	72
3.4	Breakdown of obtained video levels for each considered network trace	76
4.1	A sketch of the streaming pipeline under test.	82
4.2	Bitrate reduction (%)	86
4.3	Video Multi-Method Assessment Fusion (VMAF) as a function of the user's head yaw angle α	87
4.4	Worst case Visual Quality vs Bitrate reduction trade-off	88
4.5	Worst case Visual Quality vs Bitrate reduction trade-off for each video	90
4.6	Frame extracted from the WhiteLions360 video	91
5.1	Workflow of the TAPAS-360 tool	95

List of Tables

1.1	Additional MPD descriptors defined in Omnidirectional Media Format (OMAF)	47
1.2	Additional <i>Restricted Scheme Types</i> defined in OMAF	48
1.3	OMAF Video Profiles	49
2.1	Average bitrate reduction (with 95% confidence interval reported in the parentheses) for the considered downscaled resolutions in the case of CRF=20.	65
3.1	Parameters used to encode the video levels and corresponding encoding bitrates	74
3.2	Average Bandwidth and Standard Deviation for each considered network trace	74
3.3	Average and standard deviation percentage of the reduction of segments bitrate in the case of segments duration respectively equals to 1.6 s and 3.2 s, for each considered network trace.	75
4.1	The video catalog used in the test	84

LIST OF TABLES

Introduction: Dissertation Overview

In the last decade, we have been witness to the explosion of the Internet media delivery services, with the exponential grow of popular video streaming companies such as YouTube and Netflix. In fact, according to the report published by Cisco [6], global Internet traffic will reach 3.3 ZB (ZettaByte = 10^{24} Byte) in 2021, of which video traffic will represent more than 80%.

In the vast technological context of online video streaming, Virtual and Augmented Reality applications are becoming increasingly popular thanks to the improvements and the penetration of cheap Head Mounted Displays (HMDs) available in the consumer market. To testify the importance of such a technology, it is worth mentioning that leading platforms are already delivering Immersive videos to their users: as an example, Google has introduced an Immersive video player on Youtube; Facebook has invested over 2 billion euros, creating a 360-degree video player in collaboration with Oculus Rift; Samsung developed its Samsung Gear VR augmented reality device; Sony has introduced PlaystationVR. The ability to stream Immersive videos, or Omnidirectional Video (OV) contents, is a key enabling technology for several emerging applications such as immersive cinema, social-media, and health-care, just to name a few.

For seek of clarity, OV contents are those videos produced by capturing the whole 360 scene in all directions simultaneously with a bunch of video cameras [7]. The user, equipped with a HMD, is free to explore the recorded environment. Streaming immersive videos with an high QoE to viewers requires resolutions larger than Ultra High Definition (HD), i.e. 4K and beyond. As a proof of this, the popular Netflix streaming company recommends an internet connection bandwidth of 25Mbps for the Ultra HD

0. INTRODUCTION: DISSERTATION OVERVIEW

video streaming [8]. Nevertheless, less than 25% of the global internet connections satisfies that requirement [9].

For these reasons, the provisioning of such new services pose numerous new issues, among which we mention:

1. the standardization of new video formats;
2. the design of new adaptive streaming algorithms;
3. the design of compression techniques suitable for immersive videos.

As a proof of this, World Standard Organizations - such as the Video Coding Experts Group (VCEG) and Moving Pictures Experts Group (MPEG) - have spent a lot of efforts to create and introduce new streaming systems for immersive videos [10].

Starting from these premises, the present work wraps around the main topic of providing advanced control algorithms for Immersive streaming applications, with the aim of optimizing resources, with a particular emphasis on consumption of network bandwidth, server storage and client computing capabilities.

With the focus on bandwidth optimization aspects, a methodology for generating Immersive contents specifically designed to optimize the video bitrate consumption has been conceived and implemented. The performance indicators of the conceived optimization technique have been evaluated, in terms of bitrate reduction and resulting visual quality in function of the user's viewport. Through an extensive experimental campaign, some insights useful for the encoding of immersive videos have been catch and the best theoretical trade-off between bitrate reduction and visual quality (evaluated with both Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measurement (SSIM) visual quality metrics) in viewer side has been found.

Within the context of the Cloud-based pPlatform for Immersive adaPtive video Streaming (CLIPS) project, the architecture of a Dynamic Adaptive Streaming over HTTP (DASH)-based control system for the adaptive streaming of immersive contents has been proposed. The DASH-based system is based on two distinct control algorithms which dynamically cooperate for adapting both to the varying network conditions and to the movement of the user's viewport. The optimizing methodology described ahead has been used as a content generation algorithm. The complete streaming platform has been implemented and a performance evaluation has been carried out.

Moreover, by following the most recent developments of the State-of-the-Art in the optimization techniques for the immersive video streaming, the two techniques used

for implementing bitrate reduction of spatially partitioned immersive videos have been identified. To investigate the relationship between the obtainable bitrate reduction and the resulting video quality (evaluated with the Video Multi-Method Assessment Fusion (VMAF) visual quality metric), the identified techniques have been tested against a video dataset lasting a total of around 88 hours of immersive video contents.

Finally, the open-source TAPAS-360 tool has been developed with the aim to aid in the research community the rapid prototyping of viewport adaptive control algorithms.

Moreover, other research activities, strictly connected to those aforementioned, have been carried out during the PhD work.

To conclude, a brief description of the structure of this thesis is provided below:

- Chapter [1](#): *The Immersive Streaming technology*. It introduces the Immersive Streaming ecosystem, with a particular emphasis on the differences with respect to traditional streaming. The main features of the upcoming technology for the immersive streaming are also introduced, with a particular focus on the standardized protocols and the variety of hardware devices specifically designed for such applications.
- Chapter [2](#): *Reducing the Network Bandwidth Requirements for 360 Immersive Video Streaming*. It provides the description of an encoder-agnostic technique to reduce bandwidth requirements to stream Immersive Videos, which exploit the RoI concept.
- Chapter [3](#): *A DASH-compliant immersive streaming architecture*. In this Chapter a DASH-compliant video streaming control system for 360 immersive videos is provided, identifying the most important high-level components. As well as, the results of an extensive experimental evaluation on a proof-of-concept demonstrator are reported and discussed.
- Chapter [4](#): *Bitrate Reduction for Immersive Streaming: Comparing Variable Quantization Parameter (VQP) and Variable Resolution (VRES) Approaches*. It focuses on the most used solutions designed for implementing the tiling technique. Moreover, the results of a preliminary performance investigation are also provided and discussed.
- Chapter [5](#): *TAPAS-360: a Tool for the Design and Experimental Evaluation of 360 Video Streaming Systems*. This Chapter presents TAPAS-360, an open-source software allowing the rapid prototyping of viewport-adaptive control algo-

0. INTRODUCTION: DISSERTATION OVERVIEW

gorithms for Immersive Video Streaming. It describes thoroughly the main features and the software components of the TAPAS-360 tool, and provides several use cases in which TAPAS-360 can be effectively used.

Moreover, a complete list of produced scientific contributions will be presented immediately after this introduction.

Personal Scientific Contributions

Scientific contributions leading to publications during the PhD work are listed in what follows. They have been accepted for publication in international journals and conferences or they have been submitted and are still waiting for acceptance.

International Journals

1. L. De Cicco, S. Mascolo, V. Palmisano, and G. Ribezzo, “Reducing the network bandwidth requirements for 360 immersive video streaming,” *Internet Technology Letters*, vol. 2, no. 4, p. e118, 2019.
2. G. Ribezzo, L. De Cicco, V. Palmisano, and S. Mascolo, “A DASH 360 immersive video streaming control system,” *Internet Technology Letters*, 2019.

International Conferences

1. G. Ribezzo, G. Samela, V. Palmisano, L. De Cicco, and S. Mascolo, “A dash video streaming system for immersive contents,” in *Proc. of ACM Multimedia Systems Conference*. Amsterdam: ACM, June 2018, pp. 525–528.
2. G. Ribezzo, G. Samela, V. Palmisano, L. De Cicco, and S. Mascolo, “Reducing the network bandwidth requirements for immersive video streaming,” in *Proc. of Second International Balkan Conference on Communications and Networking*. Podgorica: IEEEComSoc, June 2018.
3. G. Ribezzo, L. De Cicco, V. Palmisano, and S. Mascolo, “Tapas-360: A tool for the design and experimental evaluation of 360 video streaming systems,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4477–4480.
4. G. Ribezzo, L. De Cicco, V. Palmisano, and S. Mascolo, “Bitrate Reduction for Omnidirectional Video Streaming: Comparing Variable Quantization Parameter and Variable Resolution Approaches” Accepted by 2021 Mediterranean Communication and Computer Networking Conference (MedComNet), 2021

0. PERSONAL SCIENTIFIC CONTRIBUTIONS

Other papers, outside the scope of the present document

1. E. Vogli, G. Ribezzo, A. Grieco and G. Boggia, "Fast network joining algorithms in industrial IEEE 802.15. 4 deployments." *Ad Hoc Networks* 69 (2018): 65-75.
2. Losciale, M., Boccadoro, P., Piro, G., Ribezzo, G., Grieco, L. A., Blefari-Melazzi, N. (2018, May). A novel ICN-based communication bus for Intelligent Transportation Systems. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)* (pp. 1-6). IEEE.

Submitted Papers waiting for revision

1. Ribezzo, G., Supporting Opportunistic Interaction on Cyber-Physical Systems with Augmented Reality.

1

The Immersive Streaming technology

In this chapter, the technological ecosystem supporting the streaming of Immersive contents is briefly introduced, along with the wide set of application domains that will benefit from its integration and development. The most important emerging standardized protocols for this technology are described in detail, along with the established relationship between the protocols and the different state of the art approaches.

1.1 Historical background

Although has recently become widespread, the idea of mimicking the real world in all available senses - augmenting the real world with illusory objects or even recreating imaginary worlds at all - is not new. Mirrors, lenses and light sources have been used for millennia in order to create virtual images in the real world. Such an example, theatres and museums in 17th Century used large plates of glass for merging reflections of objects with the real world in an illusion that became known a *Pepper's Ghost*. A more recent cinematographic experiment aimed at creating an illusory reality was when Morton Heilig patented Sensorama, a multi-sensory simulator which allowed to enjoy prerecorded film in color with augmented binaural sounds, scent, wind and vibration experiences [\[11\]](#).

However, the first truly computer-generated graphic experience can be traced back to a pioneer of Human-Computer Interaction (HCI), Ivan Sutherland. In his pioneering

1. THE IMMERSIVE STREAMING TECHNOLOGY

essay, Sutherland envisioned the concept of an *Ultimate Display* in which "the computer can control the existence of matter" [12].

In the subsequent work, Sutherland and Sproull designed and developed the primigenium prototype of HMD [13], known as *The Sword of Damocles* and portrayed in Figure 1.1.



Figure 1.1: The *The Sword of Damocles*.

The system was based on a CRT optical see-through HMD, overtaken by a mechanical tracking system. Computing was performed on a PDP-11 computer with custom graphics hardware. Later, the system was improved by replacing the cumbersome mechanical equipment with an ultrasonic tracker. Although primitive, their system combined the necessary display, tracking and computing components to provide the user with three-dimensional (3D) graphics that appearing to be overlaid on the real world.

Sutherland pioneering work assumed a giant role in teasing researchers interest into the HCI research field, but for the next couple of decades the most part of research on computer-generated reality involved military and government research labs only, rather

than academic or industrial.

Such an example, Thomas Furness and others at the Wright Pattern Air Force developed the advanced flight simulator Visually Coupled Airborne System Simulator (VCASS) [14] within the *Super-Cockpit* program. In the developed prototype the pilot wore a HMD in the aircraft cockpit [15], showing complex flight details in such a way to not overload him with too much information.

In 1971, the University of North Carolina realized the first prototype of force feedback ceiling-mounted arm, GROPE [16]. Then, Myron Krueger developed VIDEO-PLACE, a HCI system where participant's live video image were combined into a computer-generated world [17].

The real-time, multi-user and interactive simulator - Simulation NETworking (SIM-NET) - was implemented at Defense Advanced Research Projects Agency (DARPA) with the aim of training soldiers with various battlefield scenarios [18]. In 1984, the NASA start to produce a visual display named Virtual Visual Environment Display (ViVED), where a fully immersive virtual environment was created by connecting a stereoscopic monochrome HMD equipped with magnetic tracking system to a graphics computer [19].

The first industrial and academic attempts started in the second half of eighties [20] [21]. Jaron Lanier, the founder of VPL Company, first coined the term Virtual Reality (VR), starting the commercial distribution of popular *Data Gloves* and *EyePhone HMD* [1], which are shown in Figure 1.2. In 1989, the Fake Space Labs developed the BOOM [2] VR device, a box containing two small CRT monitors that can be viewed through two eye holes. In 1992, a team of graduate students at Electronic Visualization Laboratory at University of Illinois led by Dr. Carolina Cruz developed an immersive system named Cave Automatic Virtual Environment (CAVE), in which the 3D environment was recreated by means of a 3D projector and Liquid Crystal Display (LCD) shutter glasses [22].

With some of the key enabling technologies for artificial reality - such as tracking, display and interaction - well established, by the mid '90s the research community started to streamline the technology ecosystem. In [23], Heim depicted the 7 pillars of VR in *Immersion, Simulation, Artificial Reality Interaction, Telepresence, General*

¹<https://www.vrs.org.uk/virtual-reality-profiles/vpl-research.html>

²<http://www-cdr.stanford.edu/html/DesignSpace/sponsors/boom.html>

1. THE IMMERSIVE STREAMING TECHNOLOGY



Figure 1.2: VPL research Data Gloves and EyePhone.

Immersion and Network communication. During the World Wide Web (WWW) conference held in 1994 at Geneva, the Virtual Reality Modeling Language (VRML) summarized the most commonly used characteristics of 3D applications with the intent of bringing the VR on the web [24]. VRML was enhanced into the eXtensible 3D (X3D) standard [25].

An important role for the definition of such technology was given by Milgram and Kishino [20], with the concept of *Mixed Reality*, which is the merging together of real and virtual worlds, and a *Virtuality continuum* which is a taxonomy of the various ways in which the "virtual" and "real" elements can be combined together.

On the left hand, there is the *real world*, where the user sees an unmodified reality. Proceeding to the right, the *Augmented Reality (AR)* is found. AR systems aims at merging - through advanced tracking and positioning techniques - computer-generated virtual objects within the real world. These augmentations are spatially registered in 3D space and are interactive in real-time [26]. A step forward there is the *Augmented Virtuality (AV)*, where the presence of virtual objects is massive.

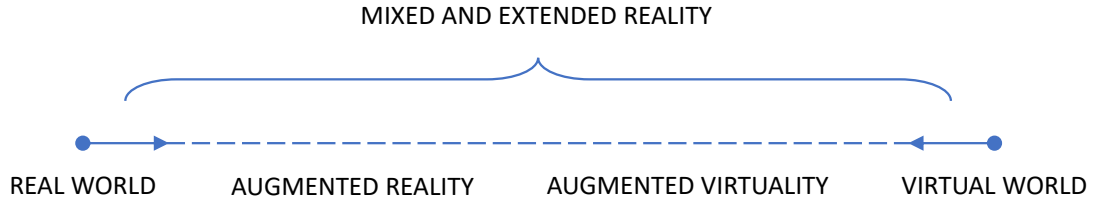


Figure 1.3: The Virtuality continuum.

Finally, the Virtual World, where a computer generates various sensory stimuli that are delivered to the human senses: stereoscopic vision, sense of hear, sense of touch and sense of smell are posed into a 3D space with the intent of mimicking an environment different from which the user is. This can be considered as a pure VR system simulating real or imaginary worlds [27].

1.2 Today's Immersive applications

Immersive multimedia for extended reality applications is becoming increasingly popular in many application fields such as, f.i., entertainment, e-learning, e-health, gaming. A high-level view is depicted in Figure 1.4.

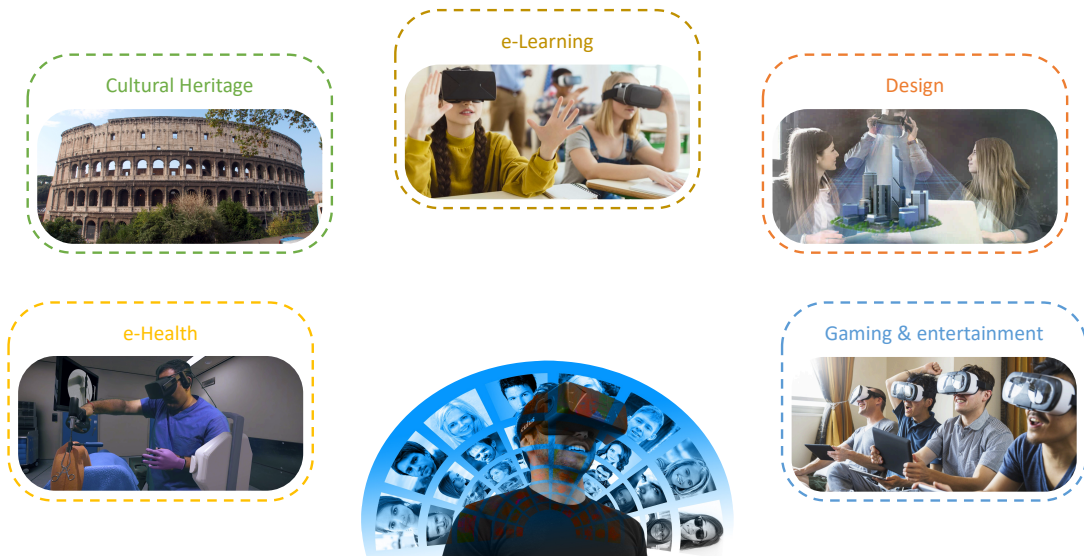


Figure 1.4: High-level view of Extended Reality applications.

1. THE IMMERSIVE STREAMING TECHNOLOGY

Concerning with e-health applications, immersive reality technologies can bring enormous benefits in training of critical non-recurrent situations. By introducing multi-user Virtual Reality in serious game, collaborative training in dynamic settings can be provided. Such an example, in [28] the authors propose EPICSAVE, which uses virtual reality with the aim of providing a practical and collaborative training simulation for paramedics. The results shown how the higher interactivity and presence given by Virtual Reality increases the learning rate. Another example was given in [29], where VR was used to enhance medical student's spatial recognition in ultrasound imagery. In [30], a VR visualization system for medical images has been developed, aiming at aiding doctors to create preoperative planning and virtual surgery. The designed system allows patient to add report of their illnesses, thus permitting doctors to create an estimate of the overall patient's condition.

Significant advancements can be provided by using VR for Cultural Heritage promotion, with completely new ways of experiencing cultural artefacts [31] [32].

A first example can be considered the ARCHEOGUIDE project [33], aiming at providing personalized tours to cultural sites through AR reconstructions and on-site information points. The enhanced reality experience is enjoyed on their own PDA screen based on real time tracking of the position and orientation in the cultural site. Moreover, the system incorporates a multimedia database of cultural material for on-line access, virtual visits, and restoration information.

Another example was the iTACITUS (Intelligent Tourism and Cultural Information through Ubiquitous Services) program [34], which aimed at increasing urban tourism by enhancing different points of interest along the city roads with overlapped information and 3D reconstruction in AR. A real experimentation of the designed system took place at Winchester Castle in Great Britain and at Villa Venaria in Italy [34].

1.3 Overview on the Streaming Protocols Ecosystem

In the early 1990s, the first video streaming systems [35] [36] were built on top of User Datagram Protocol (UDP). As well known, UDP lacks of advanced stream management mechanisms (in particular, a congestion control mechanism necessary to avoid network collapse due to congestion is completely missing): to this end, services running on top of this transport protocol need the design of such mechanisms at the application level.

1.3 Overview on the Streaming Protocols Ecosystem

With this target in mind, the Internet Engineering Task Force (IETF) started its work aimed at providing the Internet Protocol (IP) network with the capability to stream multimedia contents. The outcome of such an effort was a streaming system essentially based on three protocols: Real-Time Transport Protocol (RTP), RTP Control Protocol (RTCP) and Real-time Streaming Protocol (RTSP). The RTP [37] protocol, based on UDP, defines packet formats for exchanging audio and video contents on IP network. The RTCP [38] was a simple stream-session management protocol which enabled the connection monitoring by allowing exchange of transmission statistics. The client-server connection were established by the TCP-based RTSP [39] protocol. Figure 1.5 shows a sequence diagram illustrating the establishment of a stream session with RTSP / RTP.

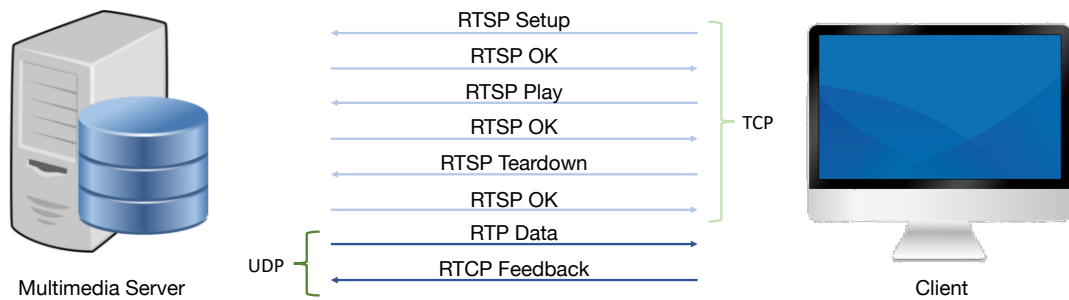


Figure 1.5: RTSP/RTP sequence diagram.

With the conceived system, IETF was able to enable streaming of multimedia contents on managed IP networks with low-overhead and low-delay delivering, ensuring stream-synchronization [40].

Nevertheless, at that time the common industrial practice was to implement proprietary technologies, protocols and control algorithms for video streaming. Furthermore, the IP network itself was sectioned into several private subnetworks, due to the architectural issues of IP protocol - which pushed the scientific community to start rethinking the design of the network [41] [42]. In this highly fragmented scenario, the provision of the streaming service was carried out by multiple proprietary Content Delivery Networks (CDNs), many of which do not provide support to RTP. In addition, router on subnetted networks and firewalls often block RTP packets on path. Finally, RTP streaming requires a separate streaming session for each client being established on server, strongly hindering its large-scale deployment.

1. THE IMMERSIVE STREAMING TECHNOLOGY

It is worth noting here that these early systems were based on the assumption that containment of end-to-end latency was a key performance index in the design of video streaming systems, a hypothesis that in the following has been proven wrong [43] [44] [45]. The designed congestion control algorithms did not implement retransmission mechanisms and, therefore, packet loss events (both due to congestion and unreliability of the transmission medium) resulted in an important degradation of the video quality.

The situation changed in 2005 when YouTube first adopted the Progressive Download Streaming (PDS) approach. With PDS, the video is downloaded like any other file via an Hyper Text Transfer Protocol (HTTP) / Transmission Control Protocol (TCP) connection using a normal Internet browser. The sequence diagram in Figure 1.6 outlines the streaming of multimedia contents in the case of PDS.

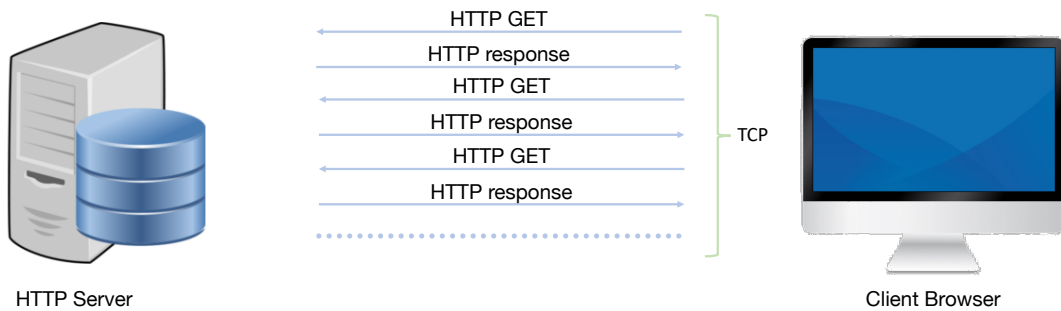


Figure 1.6: PDS sequence diagram.

As depicted in 1.6, with PDS streaming the multimedia content is fetched by progressively issuing the HTTP GET primitive. The design of PDS allowed for solving the most of the aforementioned problems: first of all, being built over top of HTTP, can pass through any firewall and private network; moreover, can be easily deployed on the existing CDNs; finally, TCP ensures a reliable transmission, thus increasing the overall video quality. This approach was improved later with the introduction of the paradigm called HTTP Adaptive Streaming (HAS) [46], which added the possibility of adapting the video bitrate based on the user device and the end-to-end band [40], thus becoming the dominant technology and implemented today by all video streaming platforms.

Among the various adaptive streaming techniques proposed in the literature, *stream-switching*, known also as Multi Bit Rate (MBR) or Adaptive Bit Rate (ABR) streaming, represents the most widely used technique today. A high-level view of this technique is showed in Figure 1.7. In a nutshell, the video content is stored on a standard HTTP

1.3 Overview on the Streaming Protocols Ecosystem

server and a client fetches the video by employing an HTTP connection. The video content is encoded at different bitrate levels which form the video levels set $L = l_1, l_2, \dots, l_M$ with $l_i < l_{i+1}$ [47], [48]. Each video level l_i is logically, or physically, divided into segments of constant duration.

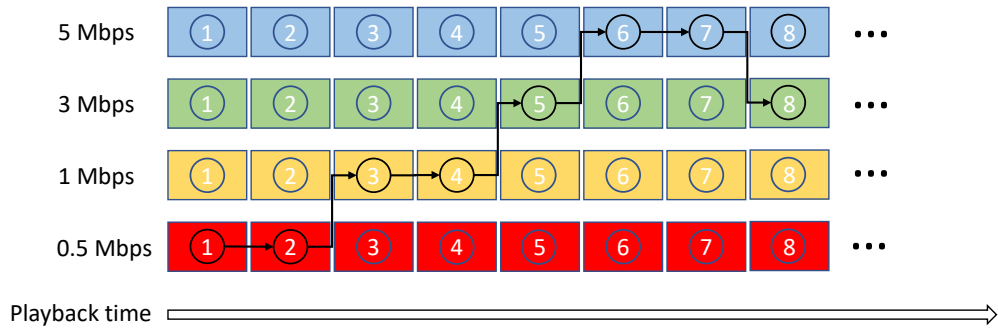


Figure 1.7: Diagram of the ABR streaming.

There are currently two standards most used industrially to achieve stream-switching: the HTTP Live Streaming (HLS) proposed by Apple [49] and the DASH [50].

1.3.1 HTTP Live Streaming

HLS [49] is the streaming protocol designed by Apple to be used for the delivery of media contents on its platforms and devices. HLS defines a format for representing audio-video streams in which multiple versions of the same stream are provided at different quality levels. Each version of the stream is coded so that it can be divided into fragments (or segments) that can have a duration that typically ranges from 4 to 10 seconds. A fragment is then encapsulated into the MPEG-2 Transport Stream (MPEG-TS) container, thus each one can be reproduced independently from the previous and subsequent ones. HLS specifications requires the list of fragments is indexed in a Manifest named *M3U8*. The *M3U8* Manifest is a file in text format and contain, one per line, the Uniform Resource Locators (URLs) of the video fragments to be downloaded jointly with the metadata (duration of the fragments, resolution, encoding bitrate, and so on) useful to player for determining which version of video to download depending on available bandwidth and screen resolution.

1.3.2 MPEG Dynamic Adaptive Streaming over HTTP

A step ahead toward the standardization process was performed when MPEG work-group developed Dynamic Adaptive Streaming over HTTP (DASH), which became an international standard in 2011, being published as *ISO/IEC 23009-1: 2012* [50]. DASH standard aims at providing efficient delivery of multimedia content through HTTP connections, also ensuring interoperability between proprietary solutions. To this end, the DASH standard was designed to reach the following goals:

1. existing technologies - such as containers, codec, Digital Right Management (DRM), and so on - can be easily reused;
2. it can be deployed on the existing CDNs;
3. it enables seamless switching of the visual quality to adapt at the varying bandwidth conditions, devices capabilities and user preferences;
4. it can coexist with existing streaming technologies.

Thanks to the aforementioned features, Dynamic Adaptive Streaming over HTTP (DASH) has become the de-facto standard employed today in the industry for dynamic and adaptive video streaming of media over HTTP [51] [52] [52] [53].

In summary, the streaming session has been designed as in the following. As usual for the MBR techniques, the video content is encoded at different bitrate and resolution level, named *Representations*. Then, each *Representation* is divided into chunk segments of constant duration. The HTTP server indexes these segments and produces a Manifest, named Media Presentation Description (MPD), which provides all the information about the segment, such as encoding bitrates, resolution, duration and a Uniform Resource Identifier (URI) allowing the client to access such described segment. The client downloads and analyzes the MPD and builds a data structure used to download video segments. During the video playback, a control algorithm running on client dynamically selects the video level to be streamed at each segment download.

1.3.2.1 The MPD Manifest

As aforementioned in the previous section, DASH systems require the multimedia content to be made available server-side in various bitrate levels, each of them splitted into several segments. Moreover, multimedia contents usually consist of several media

1.3 Overview on the Streaming Protocols Ecosystem

components (for example, audio, video, and text), each one having specific characteristics. The client, to perform bitrate adaptation, needs to know all of this information. The Media Presentation Description (MPD) is an eXtensible Markup Language (XML) document that deals with listing these data and make available to client. Figure 1.8 depicts the MPD data model.

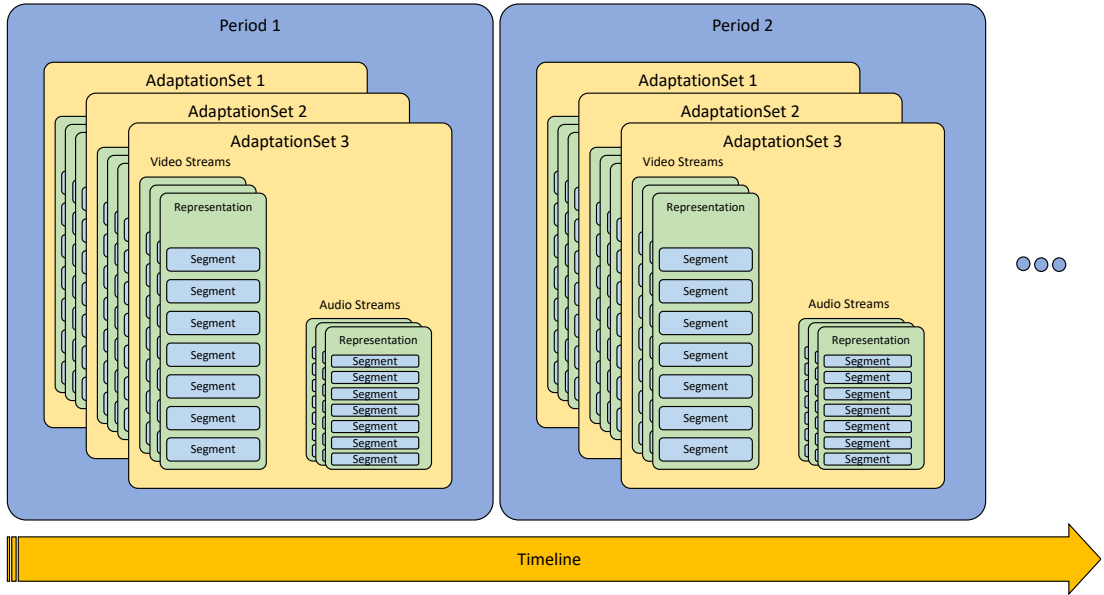


Figure 1.8: MPD Data Model.

As summarized in Figure 1.8, the MPD consists of a sequence of *Periods*, each *Period* indicating a precise interval along the temporal axis in the form of start time and duration. A *Period* is composed by one or multiple *adaptation sets*. An *Adaptation Set* contains one or multiple media streams and provides information about the type of the grouped media components. The common case is to have an adaptation set for each video / audio track compositing the same multimedia content. As an example, adaptation sets are used to differentiate streams for localization purposes, and to provide different subtitle texts. An adaptation set can includes one or multiple *Representations*. A *Representation* identifies the specific encoded alternative of the same media component. Different representations can be used to differentiate multimedia streams by bitrate, resolution, number of channels, or other characteristics. Each representation contains temporal list of *Media Segments*, each of them pointing to the specific media

1. THE IMMERSIVE STREAMING TECHNOLOGY

stream chunk. The pointer is represented by a URI. In this way, a *Segment* can be downloaded by using the simple HTTP GET or with an HTTP GET with byte ranges.

1.3.2.2 Media segments format

In the previous paragraph, the MPD Manifest has been described as a collection of metadata useful to provide a picture of the multimedia content stored on server. MPD is primarily used by DASH players 1) to strive for the best flavour of the multimedia content based on the device capabilities and 2) to dynamically adapt bitrate based on network conditions. This is possible because the multimedia content can be accessed as an ordered temporal sequence of *Media segments* encoded at different target bitrate. In DASH, a *Media segment* is defined as an HTTP addressable data structure containing one or more media samples [50], expressed by means of a URI. In particular, DASH defines different addressing modes [54]

- Indexed addressing: into this mode, a *Representation* consists of a single track, composed by an initialization segment followed by the sequence of multiple media segments. Usually, the single media segment can be accessed by HTTP GET with byte range.
- Explicit addressing: into this mode, a *Representation* consists of a set of media segments. The Representation provides a template URL, jointly with information about the initialization segment, the total number of segments, the start and the duration timestamp for each media segment. Following the given template, a client is able to access to the specific media segment by constructing an URL with the information provided and the needed playback time.
- Simple addressing: into this mode, a *Representation* consists of a set of media segments. The Representation provides a list of URLs or a template URL, jointly with information about the initialization segment, the total number of segments, the start segment and a nominal chunk duration. Differently from the *Explicit addressing* mode, no additional information about the start and the duration timestamp for each media segment is provided.

On the one hand, the information contained in the *Media Segment* data structure needs to be encoded into the MPD Manifest to be made available at client. To the purpose, DASH defines a set of specific elements to encode these pieces of information.

1.3 Overview on the Streaming Protocols Ecosystem

Specifically, the following elements has been defined to URL can be specified by using the following elements:

- the *BaseURL*, mostly used with the *Indexed addressing* mode;
- the *SegmentTemplate*, when the list of segments can be described with a URL template;
- the *SegmentList*, when a list of segments cannot be described by a common template.

Moreover, the *SegmentBase* element tag is used to define information shared between media segments. *SegmentTimeline* tag provides the start and duration timestamp for each media segment in *Explicit addressing* mode. It is worth noting here that this information can also appear at higher levels in the MPD. In this case, the information provided has to be considered as default unless overridden by *SegmentTimeline*.

On the other hand, DASH defines mechanisms to use common ISO Base Media File Format (ISOBMFF) [55], MPEG-TS [56] and Matroska - Web Media Project (WebM) [57] segment-container formats. Nevertheless, DASH is media codec agnostic and supports both multiplexed and non multiplexed encoded content.

1.3.3 MPEG Common Media Application Format

The Common Media Application Format (CMAF) [58] is an extensible standard defined by MPEG aiming at providing an unified encoding and packaging of segmented media objects for delivery and decoding on customer devices in adaptive multimedia presentations. CMAF abstracts the delivery and presentation of multimedia contents with a hypothetical application model, thus allowing a wide range of implementations including HLS and MPEG DASH. The CMAF standard specifies the usage of a subset of commonly used standardized media technologies and profiles. The CMAF specification organizes media contents into several media objects as in the following:

- CMAF Track: in CMAF, each media content - like audio, video, and subtitle - is stored in a specific ISOBMFF-based track, composed by a CMAF Header and one or more CMAF Fragments. Encoded media may optionally be encrypted with MPEG Common Encryption;
- CMAF Switching Set: the same media content can be encoded in alternative tracks using different target bitrates and resolutions;

1. THE IMMERSIVE STREAMING TECHNOLOGY

- Aligned CMAF Switching Set: alternative CMAF Switching Sets encoded from the same source with different encodings (such an example, different codec);
- CMAF Selection Set: Different Switching Sets of the same media type that may include alternative contents (for example, different languages or camera angles);
- CMAF Presentation: One or more presentation time synchronized Selection Sets.

The CMAF media organization has been designed to allow seamless switching of alternative encodings of the same content at different bit rates, frame rates and resolution. The CMAF *Hypothetical Reference Model* defines an abstract way how different tracks are delivered, combined, and synchronized in CMAF Presentations. The specific manifest and delivery protocol is left unspecified, thus enabling HLS Playlists and DASH MPD to share the same media resources and consequently allowing efficient caching even when delivering to mixed HLS-DASH platforms. Shared media resources are indicated as CMAF *Addressable Objects* and consist of:

- CMAF Header: It includes information for initializing a track.
- CMAF Segment: It contains one or more consecutive fragments belonging to the same track.
- CMAF Chunk: It is formed by a sequential subset of samples belonging to the same fragment.
- CMAF Track File: The entire ISOBMFF-based track.

The definition of an hypothetical model based on addressable media objects enables the creation of HLS Playlists and DASH MPDs on-the-fly, based on device capabilities. For these reasons, Apple was interested into its development from the beginning, thus adding ISOBMFF support to HLS streaming format [\[59\]](#).

1.3.4 Coding Standards

The encoder plays a crucial role in a streaming system: the efficient video (and audio) compression with the goal of enabling the delivering of even higher resolution contents on the best-effort network. At this time, both traditional two-dimensional (2D) videos and OVs share the same encoders, such as MPEG's Advanced Video Coding (AVC)/H.264 and HEVC/H.265, or Google's VP9, VP10, AV1, etc. The compression efficiency has enhanced significantly over the years, following the development of even more sophisticated encoders. It is worth to remark here that streaming OV has

much higher bitrate requirements compared to traditional 2D videos, up to 200-1000 Mbps [60]. Hence, improving efficiency in compression is an key research topic.

Nowadays, AVC is the most used encoder for streaming services [61]. MPEG AVC [62] were based on a 16x16 macroblock structure for frame encoding. Bitrate reduction were achieved by the motion prediction mechanisms. The next generation of MPEG encoders, named HEVC, were conceived for saving nearly 50% video bitrate compared to the AVC at the same subjective quality [61]. HEVC is considered the State-of-the-Art in coding research field. In the following, a high level description is provided, highlighting the key features of such an encoding standard.

1.3.4.1 MPEG High Efficiency Video Coding

HEVC [63] has been designed to increase the resolution of coded video and increase the use of parallel processing [61]. The high level architecture of HEVC has been designed with a layered approach aiming at coding, storing and secure transmitting the video signal.

In HEVC, the Video Coding Layer (VCL) contains the features for coding the video signal. The VCL in HEVC is based on the same hybrid approach as in AVC: intra/inter picture prediction and DCT-based transform coding. *Intra-picture prediction* is referred to as when prediction is performed on spatial data from region-to-region within the same picture. *Inter-picture prediction* involves the use of motion data (in the form of Motion Vector (MV)) and algorithms for predicting the selected picture with data coming from temporally different pictures.

Compared with fixed AVC macroblocking, HEVC uses a more flexible partitioning. Figure 1.9 shows in a nutshell the encoding steps in HEVC. As depicted in Figure 1.9, each video frame is split into several Coding Tree Units (CTUs), square or rectangular regions of 16, 32 or 64 samples in the video corresponding to the macroblock structure adopted in AVC. Smaller CTU using a tree structure and quadtree-like signaling is possible, at the cost of reduced encoding efficiency. CTUs are partitioned into one or more Coding Units (CUs), representing the basic coding block in HEVC. A CU can be further split into more Prediction Units (PUs) and Transform Units (TUs), that are logically separated blocks serving as basic units in the course of the prediction phase and transform coding phase, respectively [64].

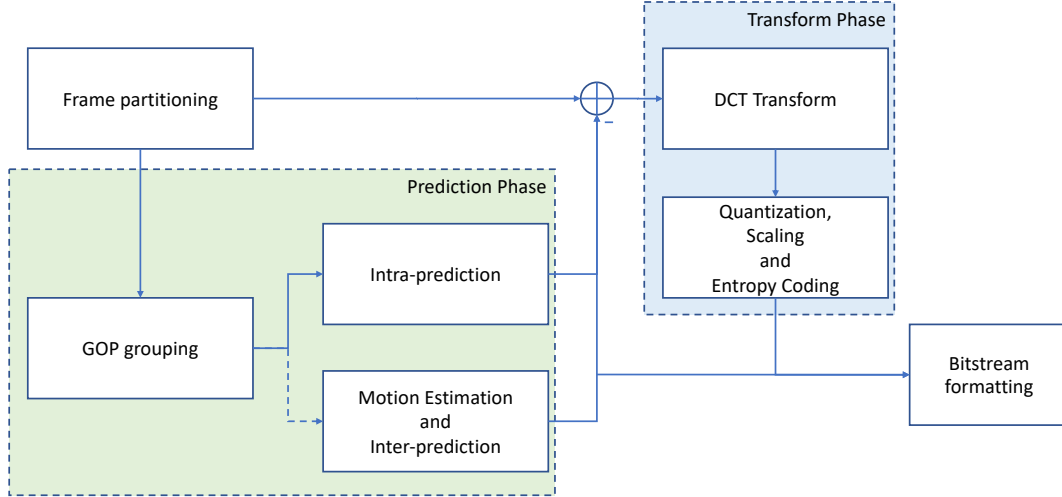


Figure 1.9: Diagram summarizing HEVC encoding.

After spatial partitioning phase, the frames - divided into several PUs - composing the video are arranged into sequences of a certain length, named Group Of Pictures (GOP), to undergo the prediction phase. The first frame in the sequence is encoded with intra-picture prediction, thus serving as clean random access point from which the decoding process can start safely [65]. The rest of frames in the GOP are inter-picture predicted by computing the MV between the PU in the current frame and the respective PU coming from the first frame in the sequence (intra-predicted frame from the following sequence can be used when bi-prediction is enable) [64].

The *residual signal* - which is the difference between the original TU block and its *intra-picture* or *inter-picture* prediction - is transformed by a Discrete Cosine Transform (DCT) spatial transform. Then, transform coefficients are scaled, quantized and entropy coded with Context Adaptive Binary Arithmetic Coding (CABAC) algorithm. Finally, both prediction and transform outcomes - forming the *VCL data* - are formatted into a valid *bitstream* in order to be stored on server or streamed.

HEVC, as its predecessor AVC, provides tools and syntax elements aiming at formatting the bitstream in segmented fashion, thus being streamed over modern packet networks. The key elements composing HEVC bitstream are the following [66]:

1. *Network Abstraction Layer (NAL)*: The NAL is the basic unit which provides the mapping for VCL data into logical data packet. In HEVC, a NAL is composed by a two-byte long header, which identifies the type NAL unit. The VCL NAL units carry coded picture data, whereas non-VCL NAL allows signalling of supplemental information required for decoding correctly the bitstream.
2. *Parameter Set (PS) structure*: PSs contain shared information between different pictures of segments in the bitstream, providing a robust mechanism for conveying data that are essential to the decoding process [66]. HEVC inherits different kind of PS from its predecessor AVC: Picture Parameter Set (PPS), which allows signalling of information at picture level; and Sequence Parameter Set (SPS), carrying out information about a GOP. In addition, HEVC introduces the Video Parameter Set (VPS) structure, with the aim of conveying information that is applicable to the entire video sequence, including the dependencies between temporal sublayers. VPSs are exploited to provide HEVC with embedded support for enhanced feature such as Scalable Video Coding (SVC), 3D and multiview video coding.
3. *Slices*: Slices are sequence of CTUs that are processed with raster scan ordering, which form a spatial partitioning of a picture aiming at enabling resynchronization in the event of data losses [61]. A slice can be decoded independently from other slices of the same picture in the sense that entropy coding, signal prediction, and residual signal reconstruction are performed only with CTUs within the slice itself, hence breaking causality in the decoding process of a frame. A slice can either be an entire picture or a region of a picture and are usually packetized into different NALs.
4. *Supplemental Enhancement Information (SEI) messages*: SEI messages provide a mechanism for signalling additional metadata that are not required for the correct decoding of video frames. SEI messages are used for transmitting optional information about the frame timing, the color space used in the video signal, frame packing information such as stereoscopic video, optional display and rendering information, and so on.

Furthermore, HEVC provides supplemental features with the goal of leveraging enhanced parallel processing capability available on modern Central Processing Units (CPUs) and Graphical Processing Units (GPUs):

1. THE IMMERSIVE STREAMING TECHNOLOGY

1. *Tiles*: Tiles is a feature introduced in HEVC for the purpose of enabling parallel processing and spatial random access to local regions of video frames. Conceptually similar to slices, Tiles are independently decodable spatial rectangular partitions of a picture, but not demands for additional headers signalling, hence reducing bitstream size for high resolved video. Unlike slices, decoding of tiles requires limited threads synchronization [67].
2. *Wavefront parallel processing*: HEVC introduced Wavefront Parallel Processing (WPP) with the aim at providing a finer level of parallelism and offering better compression performances with respect to tiles. With WPP, the CTUs belonging to the same slice are arranged into rows. The decoding process starts with CTUs in the first row, a second thread is launched for decoding of the second row as soon as the processing of two CTUs in the first row is ended, a third decoding thread is launched when the processing of two CTUs in the second row is ended, and so on. In this way, the entropy coder of each thread can infer its own context model from that used the preceding row with a two-CTU delay. Wavefront entry points allow random access to the data associated with a particular WPP.
3. *Dependent slice segments*: data associated with a wavefront entry point or a tile can be carried in separate NAL units using the dependent slice segment data structure. In this way, data contained in a dependent slice segment can be made available to a system with low delay when needed. On the one end, if data associated with a wavefront entry point are partitioned into multiple dependent slice segments, the decoding of data contained in a dependent slice segment can start as soon as at least part of the data contained in another slice segment have been decoded, i.e. the overall decoding process must respect the wavefront decoding order. on the other hand, if data associated with a tile are partitioned into multiple dependent slice segments, the decoding of data coming from a particular slice segment can start when all dependent slice segments have been retrieved, i.e. when all data associated with the tile itself are available. Dependent slice segments are used mostly when other parallel features could affect compression performance (such in the case of low-delay encoding applications).

1.4 Algorithms for the QoE estimation

As envisioned in the previous paragraphs of this work, at the transmitter the encoder compresses the original video sequence before being passed over the transmission channel. At the receiver, the decoder decompresses the sequence into a visible format for the final user [68]. During this process, distortions are introduced into the video stream that can produce visually annoying artifacts for the user. The encoder, channel, decoder and display can introduce distortions in the video sequence causing a drop in the quality of the video itself, which can be detected through *subjective* and *objective* evaluation algorithms.

1.4.1 Subjective QoE metrics

The *subjective* investigation is by far the most efficient methodology for evaluating visual quality. Nevertheless, a reliable inquiry needs being conducted on a sufficiently high number of sample users [69]. In this research field, the International Telecommunication Union (ITU) recommendation [69] has provided a standardized approach for the subjective visual quality evaluation encompassing different evaluation methods, each one suitable for a specific use case. The methods provided by the recommendation can be used in a wide range of evaluation scenarios such as selection of algorithms, video quality level of a video connection and classification of video system performance. In summary, ITU suggests three testing methods for the experimental design [69]:

- the *Absolute Category Rating (ACR)*, also known as *Single Stimulus Method* or *Mean Opinion Scores (MOS)*, demands that the test video sequences being presented to each sample user one at a time, and rated at the end of each presentation independently on a category scale. Moreover, the recommendation suggests a five-level category scale as in the following:
 1. Bad;
 2. Poor;
 3. Fair;
 4. Good;
 5. Excellent.

Nevertheless, finer scales (such as, for instance nine-level) can be also used in the case of finer evaluations. In general, ACR provides an easy and fast testing

1. THE IMMERSIVE STREAMING TECHNOLOGY

implementation: the presentation of the stimuli is natural and similar to the common use of the video systems. Thus, ACR is generally adopted a qualification test on different video systems is needed.

- the *Degradation Category Rating (DCR)*, also known as *Double Stimulus Impairment Scale* method or Differential Mean Opinion Score (DMOS), requires that the test video sequences being presented to each sample user in pair: a first stimulus used as reference, and a second one as the effective stimulus to be evaluated. The reference and the test stimulus can be presented at the sample user at the same time with the usage of two monitors or with a synchronized doubled presentation on the same visual device. Even in this case, a five-level category scale is provided:

1. Very annoying;
2. Annoying;
3. Slightly annoying;
4. Perceptible but not annoying;
5. Imperceptible.

The DCR method is used in the cases when testing the fidelity of transmission with respect to the source signal is needed.

- the *Pair Comparison method (PC)*, implies that the test sequences are presented in pairs (as for the DCR method) but in this case there is no reference stimulus. In particular, given n systems under tests (**A, B, C, etc.**), PC method demands that the systems can be combined in all the possible $n(n - 1)$ combinations, and each test pair can be evaluated in all the possible order (that is, AB and BA). The sample user is asked to express an assessment on which element in the test pair has been preferred. With respect to DCR, the PC method has higher discriminatory power, which is an important feature in the case of several systems under test are nearly equal in quality.

Other than defining different testing methodologies, the ITU recommendation provides useful hints to the experimenter about the testing conditions to be ensured to the sample user for making the evaluation reliable [69]. Importantly, it recommends the experimental session being divided into a training session and one or more evaluation sessions. During the training session, the user is presented all the instructions for carrying out the experimentation and is let to familiarize with the testbed. Moreover, the

evaluation session can be no longer than 30 mins, for the purposes of mitigating both the recency and the forgiveness effect.

The recommendation also recommends that the number of sample users required for a subjective video quality study being between 4 and 40 subjects, suggesting a number of at least 15 users. Furthermore, the sample users should not be experienced assessors or present conflict of interest in the picture quality evaluation field, because of their judgement can be not impartial.

Finally, an evaluation methodology that is gaining momentum is the *crowdsourcing*, mostly used by commercial platforms [70]. Crowdsourcing quality evaluation uses statistical methods allowing the experimenter to poll the viewers to express subjective quality evaluations at the end of a user streaming session, rather than taking part of a laboratory test [70]. This method allows much more data samples to be obtained than a normal laboratory test, at much lower cost. Nevertheless, the validity and reliability of the data samples has to be taken into account.

1.4.2 Objective QoE metrics

The goal of a *objective* quality assessment algorithm is to provide an objective quality measurement of the image that is consistent with *subjective* human evaluation, thus trying to mimic the Human Visual System (HVS) [71]. On the one hand, the subjective quality assessment methods are costly and time-consuming, albeit yielding accurate results [72]. On the other hand, objective quality assessment metrics, relying on measures and analysis of the video signal, are a better choice in the cases of a real-time quality evaluation is necessary.

A general taxonomy of the objective quality assessment algorithms is divided into the following categories [73] [74]:

- *Full-reference*: requires the complete a-priori knowledge of the reference image, thus allowing a full comparison between the distorted image and the reference image;
- *No-reference*: the assessment of image distortion must be made without any kind of reference inherent in the uncorrupted image.

In general, *no-reference* objective algorithms are fast and usually employed in real-time deployments, for instance at capture-time, but their accuracy is quite low [75].

1. THE IMMERSIVE STREAMING TECHNOLOGY

On the other hand, *full-reference* objective algorithms, even though requiring both the reference and the degraded video content, produce better quality evaluations. Indeed, many objective quality assessment metrics have been developed in the scientific literature over the years, some of them leveraging different definitions of visual quality.

In the following a brief State-of-the-Art of full-reference algorithms for the quality assessment will be provided, which have been of great interest in my research activities.

First attempts to evaluate the *image/video fidelity* [74] were essentially based on the Mean Square Error (MSE) calculation [76], as in the case of the PSNR quality metric [77]. Given an image having resolution $H \times W$ and defined $MSE = \frac{1}{HW} \sum_{i,j} (x(i,j) - y(i,j))^2$, the PSNR is simply derived as:

$$PSNR_{dB} = 10 \log_{10} \frac{S^2}{MSE} \quad (1.1)$$

The PSNR metric is generally recognized to work well in discriminating minute distortions [78]. Nevertheless, the PSNR quality metric does not perform very well in matching the visual quality [68] [79]. Then, great effort has been gone for designing better visual quality metrics that take advantage of known characteristics of the HVS [80] [74].

A further advancement into the quality assessment research field was the SSIM metric [1]. In this work, based on the insight that image distortion is highly correlated to the *structural information* from the user Field of View (FoV), the authors constructed a quality metric for measuring the structural changes in images. Figure 1.10 shows the block diagram for extracting the SSIM quality index.

In Figure 1.10, signal x and signal y are respectively the reference and the distorted images. The SSIM assessment index is then calculated as:

$$SSIM(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma \quad (1.2)$$

where $l(x, y)$, $c(x, y)$ and $s(x, y)$ are indexes comparing respectively the local luminance, the local contrast and the structural information, and α, β, γ are weighting parameters. The luminance, contrast and structural information are then expressed by the weber's law [81]

$$l(x, y) = \frac{2(1 + R)}{1 + (1 + R)^2 + \frac{C_1}{\mu_x^2}} \quad (1.3)$$

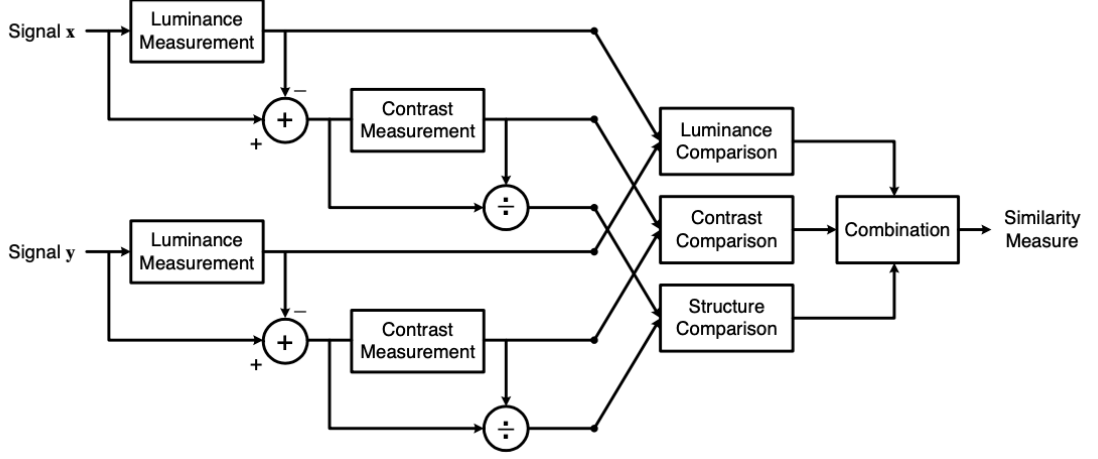


Figure 1.10: Diagram of the SSIM measurement system [1].

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (1.4)$$

$$s(x, y) = \frac{\sigma_x y + C_3}{\sigma_x\sigma_y + C_3} \quad (1.5)$$

where R is the relative luminance, C_1, C_2, C_3 are constants of integration, μ, σ are the mean and the standard deviation values for the considered signal. By assuming $\alpha = \beta = \gamma = 1$ and $C_3 = \frac{C_2}{2}$, then the SSIM metric can be formulated as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1.6)$$

The SSIM as derived before is a single-scale method. In order to facilitate its use in multi-resolution applications (such as, for instance, in MBR streaming), the authors proposed the Multi-Scale SSIM (MS-SSIM) [82]. Considering $j = 1 \dots M$ the different possible resolutions, MS-SSIM is defined as in the following:

$$MS - SSIM(x, y) = [l_M(x, y)]^{\alpha_M} \prod_{j=1}^M [c(j, x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \quad (1.7)$$

Due to the versatility of its definition, the SSIM can be easily extended for taking into account different aspects of the visual signal [83] [84] [85] [86].

1. THE IMMERSIVE STREAMING TECHNOLOGY

It is worth to remark here that PSNR- and SSIM-based quality metrics deal homogeneously with different kind of image distortions, assumption that has been demonstrated not to match with the HVS [87] [88]. Relying on the hypothesis that HVS performs multiple strategies when determining quality, in [87] the authors conceived two visual quality strategies: a detection-based strategy for high-fidelity images and an appearance-based for low-fidelity images. In [77], a distortion model, coherent with the HVS, was provided by decoupling the psychovisual effects of frequency distortion and noise injection. Following this principle, in [89] the degraded image is explicitly modeled as the combination of three components: the reference image, a linear detail loss and an additive noise impairment. The final visual quality evaluation metric is obtained by combining two different indexes: the Detail Loss Measure (DLM), which computes the loss of useful information of the test image with respect to the reference; and the Additive Impairments Measure (AIM), taking into account for information not present in the reference image, such as blurring or blocking artifacts that are commonly raised from the encoding process [89].

Even though promising, all of these evaluation techniques require a thoroughly knowledge of the HVS for extracting its behavioral model. By observing that all the aforementioned quality indexes works well in specific situations, the authors in [90] proposed to use machine-learning algorithms for fusing different quality metrics into a single index. Different machine learning algorithms were tested [91], with the SVM showing the best performances.

In summary, the quality assessment algorithm is divided in two phases: a training phase, aiming at obtaining the perceptual model associated to a given video dataset; and a testing phase, where the obtained model is used for the real video quality assessment. The training phase proceeds as follows: some elementary quality metrics are considered, and the relative score are computed on the training video dataset; then, each elementary score is normalized to match in the $[0, 1]$; the final evaluation quality metric is computed as the non-linear combination (with weighting coefficients) of the elementary scores; the weighting coefficients are the outcomes of a regressor algorithm (e.g. the SVM) by using the DMOS scores associated with the training video dataset.

After the training phase, both the reference and the distorted videos are analysed for retrieving the set of elementary metrics as done in the training phase. Then, the weighting coefficients obtained during the training phases are then used for computing

the video quality assessment. The initial algorithm has been deeply extended in the VMAF quality metric released by Netflix¹ and adopted in several of its commercial products [92] [93] [94].

The VMAF quality assessment algorithm has shown higher accuracy than conventional metrics [95] [96] [97], thus is currently considered as the State-of-the-Art algorithm in the quality evaluation research field [98] [99] [100].

1.4.2.1 A digression about the SVM algorithm

As evident, the core of the VMAF quality metric is the SVM algorithm, able at extracting the perceptual quality model. Thus, in order to make clearer how VMAF metric works, a brief description of the SVM algorithm is provided in the following.

Given the training set $(\mathbf{x}_i, y_i), i = 1 \dots M$, with $\mathbf{x}_i \in R^n, y_i \in R$ being respectively the features set and the ground truth values, the SVM algorithm task is to find the vector $\mathbf{w} \in R^n$ which satisfies the following convex optimization problem [101] [102] [103]:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & \|f(\mathbf{x}_i) - y_i\|_1 \leq \epsilon, \forall i = 1 \dots M \end{aligned} \tag{1.8}$$

where $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$ is an affine function of the given vectors \mathbf{x}_i and $\epsilon \geq 0$ is the approximation error. It is worth to remark here that $\|\bullet\|_1$ and $\|\bullet\|_2$ are respectively the l_1 and the l_2 norms.

For coping with unfeasible constraints, it is possible to introduce two slacks variables $s_i \geq 0, \hat{s}_i \geq 0$. Then, the aforementioned problem [1.8] can be rewritten as:

$$\begin{aligned} \text{minimize}_{s_i \geq 0, \hat{s}_i \geq 0} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^M (s_i + \hat{s}_i) \\ \text{s.t.} \quad & y_i \leq f(\mathbf{x}_i) + \epsilon + s_i \\ & y_i \geq f(\mathbf{x}_i) - \epsilon - \hat{s}_i, \forall i = 1 \dots M \end{aligned} \tag{1.9}$$

¹Code available at <https://github.com/Netflix/vmaf>

1. THE IMMERSIVE STREAMING TECHNOLOGY

where $C > 0$ is the penalty term of the error term [104]. Then, writing the Lagrangian:

$$L = \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^M (s_i + \hat{s}_i) + \sum_{i=1}^M a_i (y_i - \mathbf{w}^T \mathbf{x}_i - b - \epsilon - s_i) + \sum_{i=1}^M \hat{a}_i (-y_i + \mathbf{w}^T \mathbf{x}_i + b - \epsilon - \hat{s}_i) \quad (1.10)$$

where a_i, \hat{a}_i are the lagrangian multipliers, it is possible to go through a simpler problem formulation by means of the Lagrangian dual problem [101]:

$$\begin{aligned} \text{maximize}_{a_i \geq 0, \hat{a}_i \geq 0} \quad & -\frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M (a_i - \hat{a}_i)(a_j - \hat{a}_j) \mathbf{x}_i \mathbf{x}_j \\ & - \epsilon \sum_{i=1}^M (a_i + \hat{a}_i) + \sum_{i=1}^M (a_i - \hat{a}_i) y_i \\ \text{s.t.} \quad & \sum_{i=1}^M (a_i - \hat{a}_i) = 0 \\ & \hat{a}_i \leq C, \forall i = 1 \dots M \end{aligned} \quad (1.11)$$

After solving [1.11], it is possible to apply the Karush-Kuhn-Tucker (KKT) optimality conditions [101]:

$$\begin{aligned} a_i (\epsilon + s_i + \mathbf{w}^T \mathbf{x}_i + b - y_i) &= 0 \\ \hat{a}_i (\epsilon + \hat{s}_i - \mathbf{w}^T \mathbf{x}_i - b + y_i) &= 0 \\ s_i (C - a_i) &= 0 \\ \hat{s}_i (C - \hat{a}_i) &= 0 \end{aligned} \quad (1.12)$$

which allow to express b in the form:

$$b = \begin{cases} y_i - \epsilon - \mathbf{w}^T \mathbf{x}_i & 0 < a_i < C, \\ y_i + \epsilon - \mathbf{w}^T \mathbf{x}_i & 0 < \hat{a}_i < C. \end{cases} \quad (1.13)$$

Finally, the final solution of the primal problem can be found as:

$$\begin{aligned} \bar{\mathbf{w}} &= \sum_{i=1}^M (a_i - \hat{a}_i) \mathbf{x}_i \\ f(\mathbf{x}) &= \bar{\mathbf{w}}^T \mathbf{x} + b = \sum_{i=1}^M (\hat{a}_i - a_i) \mathbf{x}_i^T \mathbf{x} + b \end{aligned} \quad (1.14)$$

Then, the obtained weight \bar{w} is the best coefficient which linearly approximates the given set $(\mathbf{x}_i, y_i), i = 1 \dots M$. In the case of $(\mathbf{x}_i, y_i), i = 1 \dots M$ is a non linearly dependent set, it is possible to consider a proper feature-space transformation ϕ such that $f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$ is an affine function in \mathbf{x} . Thus it results that $\bar{\mathbf{w}} = \sum_{i=1}^M (a_i - \hat{a}_i) \phi(\mathbf{x}_i)$ and:

$$f(\mathbf{x}) = \sum_{i=1}^M (\hat{a}_i - a_i) \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b = \sum_{i=1}^M (\hat{a}_i - a_i) Ker(\mathbf{x}_i, \mathbf{x}) + b \quad (1.15)$$

where $Ker(\mathbf{x}_i, \mathbf{x}) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x})$ is a kernel function. Then, exploiting the Marcel's theorem [105], is it possible to use some proper kernel functions without the exact knowledge of the ϕ function. Generally, there are some widely used kernel functions:

- Linear: $Ker(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$;
- Polynomial: $Ker(\mathbf{x}_i, \mathbf{x}_j) = (\lambda \mathbf{x}_i^T \mathbf{x}_j + r)^d, \lambda > 0$;
- Radial Basis Function (RBF): $Ker(\mathbf{x}_i, \mathbf{x}_j) = \exp^{-\lambda \|\mathbf{x}_i - \mathbf{x}_j\|_2^2}, \lambda > 0$;
- Sigmoid: $Ker(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\lambda \mathbf{x}_i^T \mathbf{x}_j + r), \lambda > 0$;

with λ, r, d kernel parameter to be tuned properly.

1.5 Technologies for Immersive Video Streaming

Internet media delivery has evolved from being a fragmented ecosystem populated with many non-interoperable technologies to a very mature and standardized field at the base of popular on-line services such as YouTube, Netflix, etc. A fundamental feature, which has contributed to making video streaming delivery systems successful, is the possibility to adapt in real-time to end-to-end network bandwidth variations by dynamically switching between several representations of the same video content encoded at different bitrates. As thoroughly analysed in Section 1.3, such a feature is implemented by ABR algorithms. Compared to classical 2D adaptive video streaming, immersive videos add several dimensions to the adaptation, thus requiring new algorithms and software components to be designed. Immersive videos add several dimensions to the classical 2D videos and allow user to explore a scene from different point of views, enhancing the overall viewing experience. A further evolution of such systems is required to stream immersive video content which comprise three Degrees of Freedom (3-DoF) OV, or 360 videos, and volumetric content, or six Degrees of Freedom (6-DoF) videos. Specifically, Omnidirectional Video (OV), also known as 360, immersive or panoramic

1. THE IMMERSIVE STREAMING TECHNOLOGY

video, is a video format that enable the viewer to freely explore the entire recorded environment. The OV is produced by capturing a scene in all directions simultaneously with a bunch of video cameras [7]. As illustrated in Figure 1.11, each incoming video stream demands being *stitched* [106] together in such a way to recompose the entire 360 panorama.



Figure 1.11: *Stitching* process of 360 panorama.

Courtesy of <http://www.kscottz.com/fish-eye-lens-dewarping-and-panorama-stiching/>

The typical way a viewer can interact with the OV is through an HMD: in this way, the user will be able to freely explore the video moving his/her head in various directions. An industrial list of customer HMDs includes Samsung Gear VR [1], Oculus Rift [2], HTC VIVE [3], Google Cardboard [4], Google Daydream [5], and PlayStation VR [6].

¹<http://www.samsung.com/global/galaxy/gear-vr>

²<http://www.oculus.com/rift/>

³<https://www.vive.com/>

⁴<https://www.google.com/get/cardboard/>

⁵<https://www.google.com/get/daydream/>

⁶<https://www.playstation.com/en-ca/explore/playstation-vr/>

1.5 Technologies for Immersive Video Streaming

Moreover, HMD can be *standalone/mobile*, i.e. the computational engine is embedded into the HMD itself, thus allowing complete freedom of movement; or *tethered*, that requires an external computational device (such as a PC) linked by wires [40].

By the way, an even less typical way of enjoying OV is by means of the flat screen of a workstation or a smartphone, then the user can interact with the immersive video by using, respectively, the mouse (even keystrokes) or the touchscreen.

In any case, common experience suggests that the human FoV is limited. The technological outstanding is that common VR players show only a limited portion of the OV content to the viewer. This portion, named *viewport*, reaches approximately 100 degrees on the most modern HMD¹.

One of today's industrial challenges for VR device manufacturers is to be able to increase the viewing angle offered by their devices in an attempt to approach wider FoVs. A further challenge for manufacturers lies in the need to provide screens with higher resolution: the display sizes joined with the distances *eye-display* involved in a typical HMD would demand a pixel density in the order of a thousand pixel per inch (ppi) for making pixels indistinguishable [107]. Such an example, Oculus Rift is equipped with a 7 inches display positioned approximately at 3 inches from eye, reaching near 600 ppi [108]. However, resolution is a constrained resource on VR devices: maximum decoding capability is limited to 4K on most modern HMDs [109].

Anyway, streaming high QoE OV requires even higher video resolution (i.e., larger than 4K), asking for higher network bandwidth. In particular, to quantify the impact of the last issue, in [60] authors show that streaming a 360 video requires a network bandwidth of about 400 Mbps to deliver a video quality similar to that of a fullHD resolution 2D video. Enable the optimal fruition of 360 video is a particularly challenging task, due to the joint higher QoE / higher bandwidth requirements.

Therefore, great scientific effort has been spent aiming at optimizing OV content production. In the following sections, an extensive State-of-the-Art on the OV content production will be provided. Moreover, further optimization techniques will be introduced.

¹<https://virtualrealitytimes.com/2017/03/06/chart-fov-field-of-view-vr-headsets/>

1.5.1 OV projection formats

Even though OV is spherical and therefore 3D by nature (in the sense that it is originally filmed through a number of cameras oriented in different directions to cover the entire 360 view), it must necessarily be mapped on a 2D plane to make them compatible with traditional encoders and decoders. By the way, standard video encoders were conceived for 2D contents: motion estimation and compensation algorithms were designed for translational movements on rectangular blocks, thus are not optimized for compressing spherical videos [110]. During the years, a vast literature of *sphere-to-plane* mappings has been supplied striving for optimizing the encoding of OV in 2D. A taxonomy having academic acceptance [111] [112] [2] classifies the various *sphere-to-plane* techniques into essentially two different categories:

1. *Uniform Quality* or *Viewport independent* mappings;
2. *Variable Quality* or *Viewport dependent* mappings.

In this section a description of the main sphere-to-plane mapping techniques proposed in literature for the projection of OV is provided, arguing for each of them the most potentially interesting features and drawbacks.

1.5.1.1 Viewport independent mappings

The key point of this kind of *sphere-to-plane* mapping is that the visual quality is uniform for all the projected panorama. In other terms, there is no particular direction pointing to a region of the 360 panorama having higher quality with respect to the others. The most commonly used *viewport independent sphere-to-plane* mappings are:

- *Equirectangular Projection (ERP)*

The ERP format derives from cartographical techniques to generate 2D world maps. Figure 1.12 illustrates in a nutshell the process used to map a 360 scene in ERP format.

As shown in Figure 1.12, ERP aims at unfolding the spherical surface to a rectangular plane. To the purpose, the generic position of a point on the spherical surface is expressed by the angular coordinates (θ, ϕ) , respectively equal to *altitude* and *azimuth*, with $\theta \in [-\pi, \pi]$ and $\phi \in [-\pi/2, \pi/2]$. Then, the rectangular video frame is constructed by assuming the following equality $(x, y) = (\theta, \phi)$. The

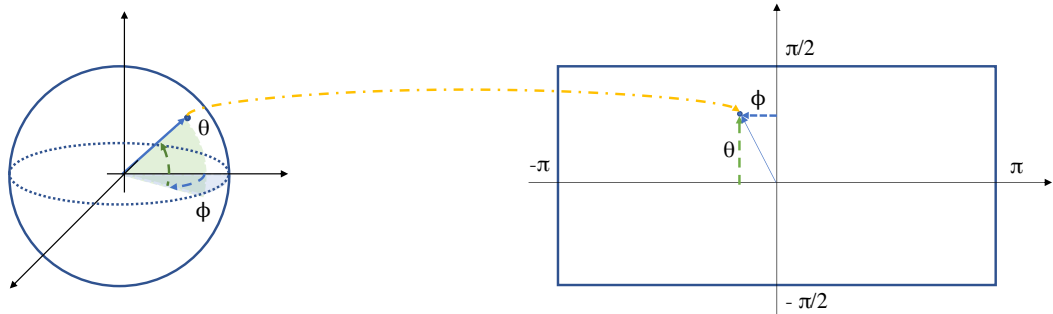


Figure 1.12: ERP sphere-to-plane mapping

equality between spherical and linear coordinates induces ERP videos to have an *aspect ratio* of 2:1.



Figure 1.13: Omnidirectional scene in ERP format [2].

Requiring minimal modification to existing 360 cameras, the ERP is by far the most common projection format for immersive videos [112]. Figure 1.13 shows the shot of a 360 scene. As put in evidence in Figure 1.13, the unfolding process leads to redundant pixels, especially at the pole areas [2], resulting in a highly distorted video. Moreover, encoding process of such over-sampled areas generates a significant waste of bitrate [111].

A slightly variant is the Equi Angular Projection (EAP) [113], which is depicted in Figure 1.14. EAP attempts to smoothly reduce the sampling rate in the y



Figure 1.14: Omnidirectional scene in EAP format [2]. Please note as straight lines are curved.

coordinate by multiplying $\cos(\theta)$. Nevertheless, the non-linear transformation function increases the shape distortion [2]: glaring at Figure 1.14 is possible to note how EAP mapping transforms straight lines into curved lines. As stated in Section 1.5.1, this issue lowers the performances in standard encoding pipeline, resulting in image quality degradation [110].

- *Cubemap Projection (CMP)*

Known as *skybox*, the CMP format has been extensively employed for texturing background in computer graphic applications [114]. Figure 1.15 summarizes the pipeline used for producing OVs in CMP format.

As pointed out in Figure 1.15, the CMP requires the spherical surface being projected onto the surface a cube, then each face of the cube is unwrapped and rearranged in a rectangular layout. Cube face rearrangement process raises discontinuities near the edges of the faces, causing losses of efficiency in terms of compression [115]. For the purpose of reducing this encoding inefficiencies, Joint Video Experts Team (JVET) recommends the use of a specific layout, as shown in Figure 1.16.

Compared with ERP, CMP presents a more uniform pixel distribution, attenu-

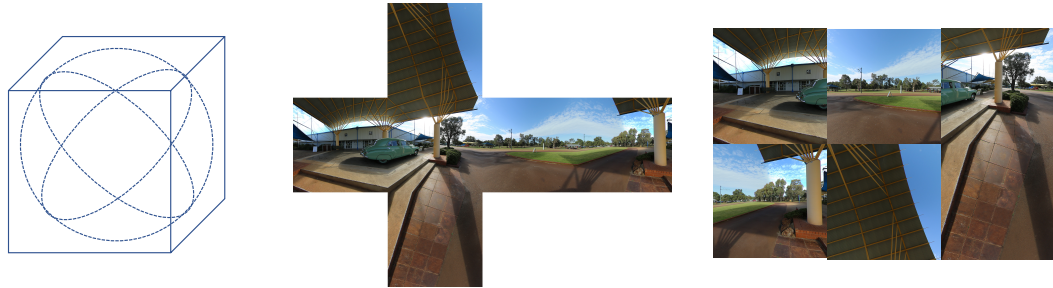


Figure 1.15: Pipeline for generating OV contents in CMP format.



Figure 1.16: JVET CMP layout.

1. THE IMMERSIVE STREAMING TECHNOLOGY

ating the oversampling issue at pole areas. Indeed, CMP needs the 25% of pixel lesser than ERP at the same visual quality, resulting in lower bitrate requirements [116]. For this reason, leading internet companies such as Facebook and Google are currently adopting this approach into their 360 products [1]. Nevertheless, discontinuities at face boundaries cannot be removed at all, causing artifacts during the rendering process [117]. A commonly adopted solution to this problem is to add extra pixels to the borders [2], increasing the final bitrate. Another drawback is the oversampling problem inside face edges [118].

- *Patch-based projection*

The main drawback of the CMP sphere-to-plane mapping is that it introduces oversampling within the cube faces [118]. Figure 1.17 illustrates a two-dimensional view of the CMP sphere-to-map process.

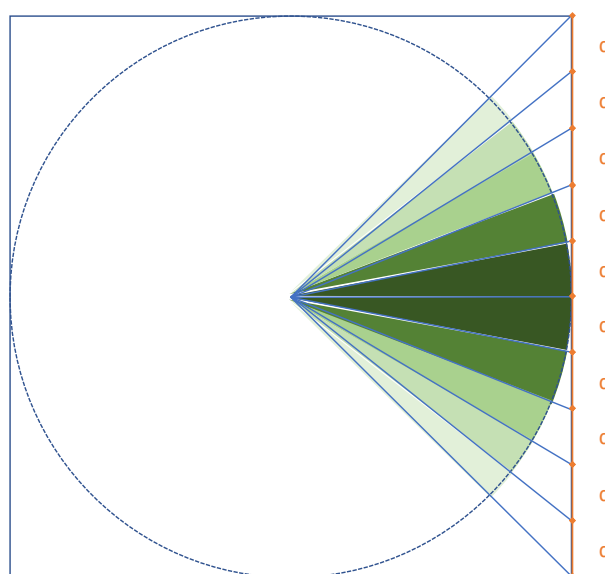


Figure 1.17: Two-dimensional illustration of the CMP projection process

As made clear in Figure 1.17, equal areas on the cube face correspond to different dihedral angles, decreasing towards face boundaries. In an effort for solving this issue, a possible solution analyzed in literature was to take advantage from using more complex polyedra having a greater number of faces - named as *patches* in

¹<https://www.youtube.com/watch?v=hNAbQYU0wpg>

²<http://paulbourke.net/miscellaneous/cubemaps/>

[2] - in such a way to reduce the introduced projecting distortion. Leveraging this idea, the author in [119] proposes the *Rhombic Dodecahedron* layout. Figure 1.18 depicts the conceived *sphere-to-plane* workflow.

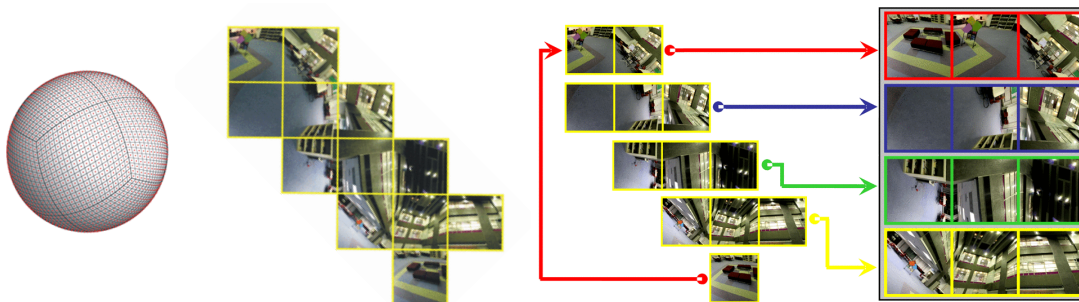


Figure 1.18: *Rhombic Dodecahedron* video projection workflow.

As detailed in Figure 1.18, the spherical surface is mapped onto a rhombus dodecahedron before being splitted and rearranged into a 3×4 rectangle. Each dodecahedron side is accurately rearranged to keep the number of edge discontinuity at minimum. In the following research activity [120] [121] [122] different experts have proposed various polyhedrons and layouts in order to find the best trade-off between number of patches (which lowers the bitrate requirements due to oversampling) and number of discontinuous edges (which degrades the resulting visual quality).

Similar works [118] [123], carried out respectively by Google and Qualcomm Inc., explore a different approach that allows to solve the oversampling problem in CMP avoiding to increase the edge discontinuities.

The proposed techniques, named Equi Angular Cubemap (EAC) in [118] and Adjusted Cubemap Projection (ACP) in [123], address such a problem by adding a nonlinear transformation in cascade to the usual CMP projection. Figure 1.19 shows a graph representing the distortion between two faces in CMP.

Going into details, Figure 1.19 plots the first derivative of the sampling rate between the angular values $[0, 1]$ in radians. Orange, green and blue lines represent respectively the distortion measured after CMP projection, the linearizing function added and the final result.

- *Segmented Sphere Projection (SSP) and Barrel layout*

It's another class of sphere-to-map mappings. Basically, the SSP aims at avoid-

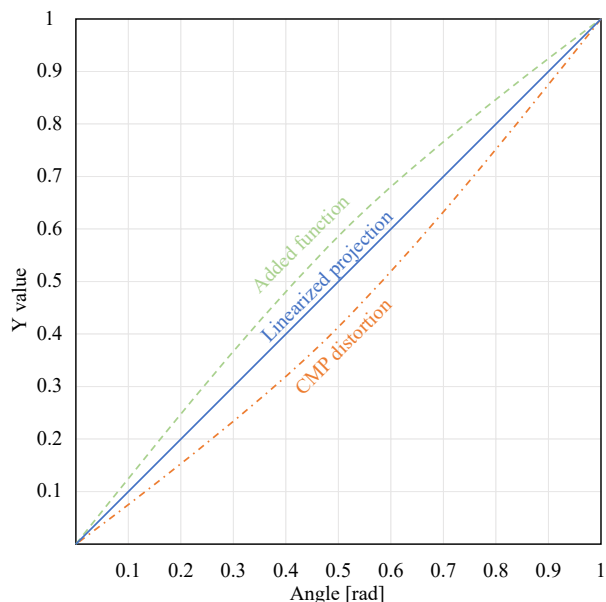


Figure 1.19: EAC/ACP linearizing function.

ing the oversampling problem in ERP by segmenting the projected sphere into different horizontal stripes, then resizing them accordingly striving for keeping the sampling rate across the stripes uniform [124]. Different stripes number are allowed to reach the desired sample rate [125]. Figure 1.20 summarizes the described approach with 5 stripes.

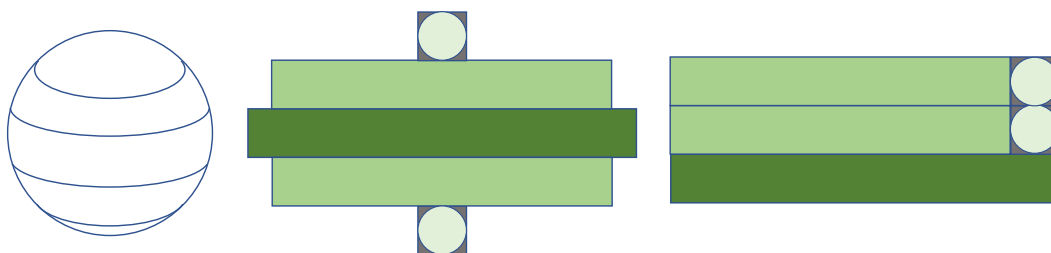


Figure 1.20: Segmentation approach.

The author in [125] further enhanced the ERP segmentation scheme into the JVET proposal [126]. As shown in Figure 1.21, JVET recommends to divide the sphere into three stripes to keep the number of discontinuity edges at minimum, jointly with the usage of the vertical layout for the sake of reducing the line

buffer.

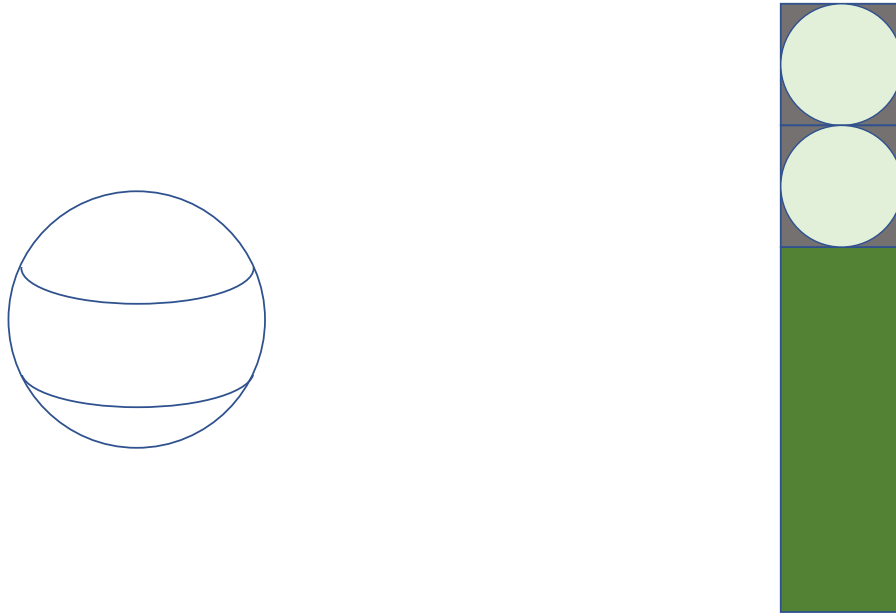


Figure 1.21: The SSP mapping proposed by JVET.

A further enhancement has been proposed by Facebook in [3] with the *barrel layout*. As shown in Figure 1.22, the *barrel layout* consists into manipulating the standard ERP in such a way to produce a pseudo-cylindrical projection. This is obtained by cropping an area of around 25% from the top and the bottom of the ERP video. The central area is vertically stretched to increase the pixel density with the aim for increasing the visual quality within areas having higher probability to be seen by viewers [3]. The top and bottom areas are reprojected to form the up and down sides of a cylinder.

1.5.1.2 Viewport dependent mappings

The idea behind *viewport dependent* techniques is to allocate more pixels to the regions of the OV most relevant for the viewer (i.e. the *viewport*), thus enhancing the quality for a specific FoV in the 360 panorama. Rendering of such a OV at client requires the able to switch between several versions of the same content, each of them showing higher quality for a specific viewing direction [2]. Examples of *viewport dependent* mapping are:



Figure 1.22: Barrel layout of a 360 video [3].

- *Pyramid projection*

The Facebook *Pyramid projection* [4] was the pioneering work which raised interest on the *viewport dependent* sphere-to-plane techniques. Figure 1.23 illustrates the pipeline used for producing OV content with the *Pyramid projection*.

Based on the same premises of the *patch-based* techniques, the idea was to project the sphere on a regular pyramid - a polyhedron formed by a square base and the apex directly above the centroid of the base, as shown in Figure 1.23. Then, the pyramid is unfolded and its sides rearranged in a rectangular layout as usual in patch-based methods. The square base - possessing the greatest amount of pixels in the projected rectangle - will correspond to the *viewport*, while the regions outside *viewport* will be mapped on the triangular sides, progressively reserving fewer pixels when moving away from it. In this way, only the viewport will encompass a full resolution video content, reducing the bitrate requirement by 80% respect to ERP [4]. Nevertheless, several versions of the same 360 scene need to be produced in order to enact users for switching between the various high quality viewports. In [4], Facebook engineers propose to cover the panorama with 30 different viewing direction, skipped by 30 degree.

The main limits of the *Pyramid projection* sphere-to-plane mapping are both implementation and performance. First of all the quality drops quickly when the user moves out of the high quality area [111]. Furthermore, hardware devices such as CPUs and GPUs do not provide support for the *Pyramid Projection*,

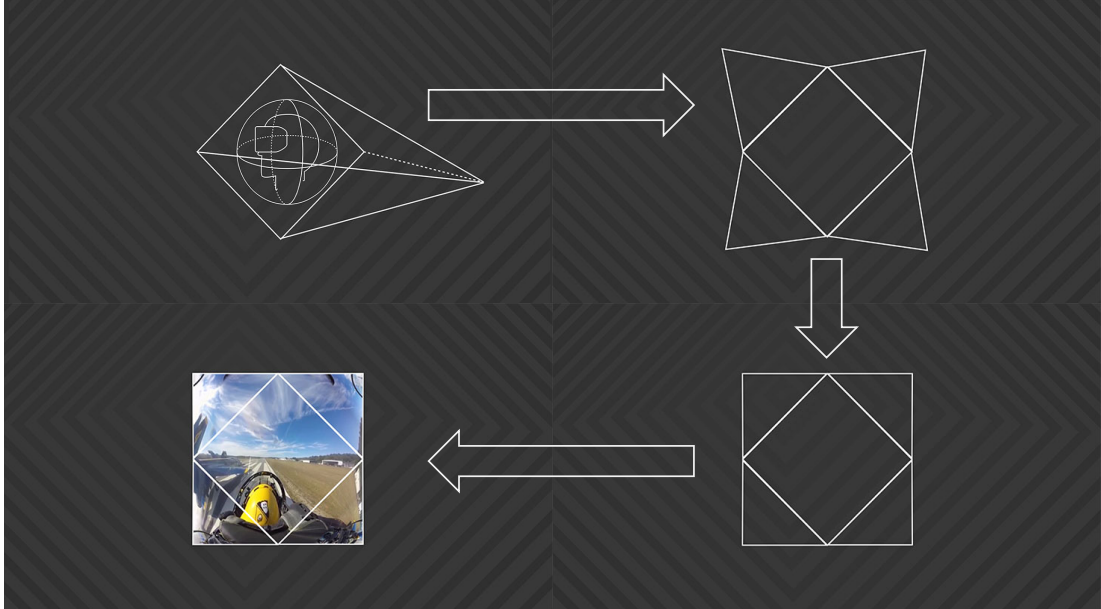


Figure 1.23: *Pyramid projection* video production pipeline [4].

making the decoding and the rendering phases computationally expensive [111].

- *Offset-cubemap projection*

It is a variant of CMP that introduces an offset to the camera that consequently determines a variable quality mapping [127]. Figure 1.24 allows to easily compare the techniques, shown respectively CMP on the left side and *Offset Cubemap* on the right side.

As depicted in Figure 1.24, the viewer is moved back from the center of the cube, thus having a larger FoV in the front direction and a smaller FoV in the opposite direction. After projection, the effect is that more pixels are allocated for the front direction. At play time, the offset is removed and such a video content is rendered as a standard CMP. The overall result is that the visual quality is emphasized in the front direction [127]. Moreover, being essentially a CMP map, the *Offset Cubemap* has not any additional computational cost.

It is worth noting here that offset-cubemap technique is considered the *State-of-the-Art* in *viewport dependent* techniques [5].

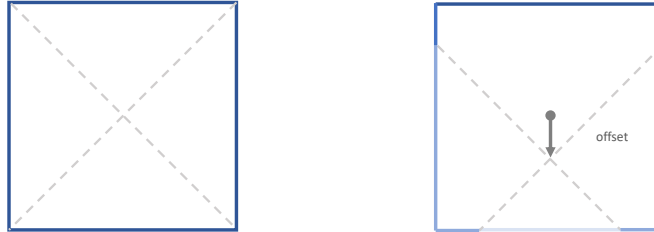


Figure 1.24: *Offset Cubemap Projection* [5].

1.5.2 Tiled Streaming

Nowadays, we are witnessing to the increasing demand for innovative video applications - for instance, interactive pan and zoom features on classical 2D videos, or Immersive and Extended Reality applications - being streamed online. The provisioning of such video applications require the usage of ultra-high resolution video (e.g. 4K, 8K and beyond). However, the streaming of such high resolution videos raises problems due to higher bandwidth requirements [60]. Also, limited hardware capabilities on constrained mobile devices may be unable to handle such ultra-high resolutions. In an effort to reduce such issues, the academic and industrial research has proposed the *tiled streaming* approach, in combination with the usage of common MBR techniques.

Tiled streaming - often referred as *Tiling* or RoI-based streaming - is a technique enabling spatial partitioning of a video into independently decodable video streams [128]. Such spatial partitioning consists of a regular $M \times N$ grid being applied to each frame of the entire video, where M is the number of columns and N is the number of rows. The traditional *monolithic* video encoding corresponds to a 1×1 tile grid. As depicted in Figure 1.25, players are allowed to download the specific *tile set* according to user RoI.

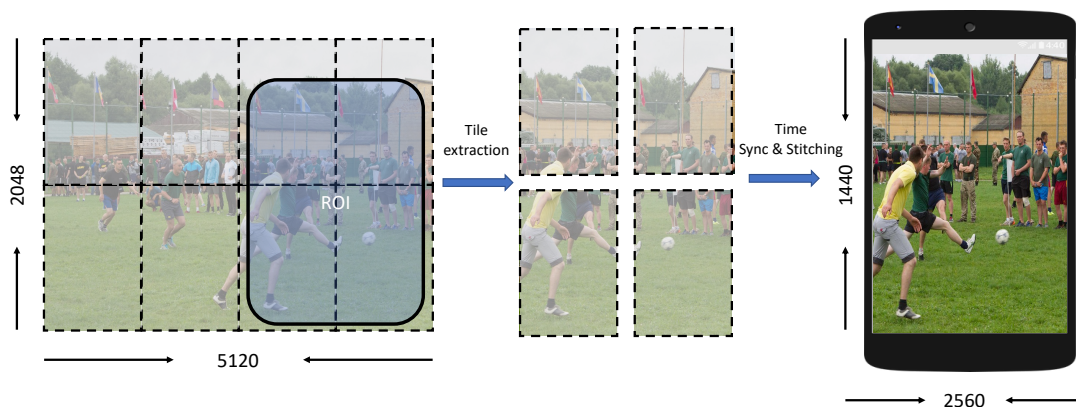


Figure 1.25: Example diagram of the Tiled Streaming approach.

The *tiled streaming* objective is to add interactivity to video players, making them able to maximize the quality of the reconstructed RoI and minimize the total bandwidth claimed by the fetched tiles.

In order to produce independently decodable video contents, no dependency between tiles is required: the result is that the client needs an ad-hoc algorithm to manage temporal synchronization between tiles and global encoding rate. Another limitation is that tiling encoding efficiency decreases when the number of tiles increases [129]. Finally, when sudden changes of the viewpoint occur, video segments of new tiles should be quickly downloaded and rebuffering events might occur in the case those segments are not downloaded fast enough.

Typically, tiling is performed within an MBR scheme, where video contents at lower resolutions can help to reduce the encoding inefficiency introduced with tiled content. If a tile at lower resolution is small enough (such in the case of thumbnails), the bitrate overhead associated with an higher resolution layer could be affordable. Moreover, the multi-resolution tiling scheme allows to increase the perceived quality in scenarios where a user-defined zooming factor is used.

This promising approach has attracted a lot of research interest: various internet standards has been extended in such a way for making them able to stream tiled content. DASH SRD feature of the second amendment of DASH standard [130] was specifically designed for providing such a technique. The feature extends the MPD allowing to describe the spatial relationships between associated tiles composing the video content.

1. THE IMMERSIVE STREAMING TECHNOLOGY

This enables the DASH client to select and retrieve only those video streams at those resolutions that are relevant to the user experience.

Moreover, also MPEG HEVC was conceived with *tiling* in mind. Tiling technique is exploited here for enabling parallel decoding with single decoder instance. Finally, MPEG OMAF enables the delivering of *tiled* OV contents in an optimized way.

1.5.2.1 MPEG DASH Spatial Relationship Description

Obeying the design principles as DASH, the SRD feature allows to specify the spatial information related to a set of multimedia contents. DASH clients uses this spatial information to determine the best set of media contents to fetch for providing a specific QoE to the viewer or to provide user interactions.

In SRD, each entire video content is described as a grid of video tiles. The reference space of such tile grid is given as a 2D coordinate system. The position of each tile in the tile grid is expressed by means of the common x , y , $width$, $height$ attributes, respectively the x , y coordinates of the top-left corner, the width and the height of the described tile. It should be emphasized here that the coordinate system is completely arbitrary and not coincide with the rendering coordinate system. In this way, complex spatial relationships can be easily described. On the one hand, explicit grid positioning (e.g. placing media in an $N \times M$ grid) is fully supported. On the other hand, the one-to-one positioning with directions allows overlapping tiles. This feature is particularly useful in Immersive streaming [7] [131]. However, SRD is codec-agnostic and allows to specify the spatial relationship for any kind of multimedia contents. Both spatial video and audio are supported [132].

In the course of the standardization effort, SRD was designed to provide a flexible instrument for describing spatial relationship between any kind of media. In particular, the SRD standard describes how multimedia contents spatially relate to each other, without specifying how a player shall use this information. In this way different composition and adaptation logics can be implemented at client-side, based on user behaviours or device characteristics. Such an example, given an MPD describing two spatially related videos, a player (possibly running on an HMD) may decide to render both videos while another player (running on desktop) could decide to play them sequentially. Furthermore, SRD describes the spatial relationship between different media contents from a content-creator perspective. The different tiles composing the

video grid may originate from the single camera shooting the entire scene, which is then splitted into several video tracks, or there may be the composition of multiple cameras each shooting a different part of the scene.

SRD Syntax

As stated in the previous Section, SRD extends the DASH standard allowing to specify the spatial relationship between any kind of multimedia contents. The spatial relationship is encoded in the MPD Manifest leveraging the *Essential Property* and *Supplemental Property* generic descriptors. Figure 1.26 shows an extract from a MPD Manifest describing two video tracks. Some attributes and elements have been omitted for brevity.

```

<Period>
  <AdaptationSet>
    <EssentialProperty schemeIdUri="urn:mpeg:dash:srd:2014"
value="0,0,0,5760,3240,5760,3240"/>
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
    <Representation id="1" width="3840" height="2160" ...>
      <BaseURL>full.mp4</BaseURL>
    </Representation>
  </AdaptationSet>
  <AdaptationSet>
    <SupplementalProperty schemeIdUri="urn:mpeg:dash:srd:2014"
value="0,1920,1080,1920,1080,5760,3240"/>
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="supplementary"/>
    <Representation id="2" width="1920" height="1080" ...>
      <BaseURL>part.mp4</BaseURL>
    </Representation>
  </AdaptationSet>
  ...
  ...
  ...
</Period>

```

Figure 1.26: A sketch of the MPD extended by SRD.

As depicted in Figure 1.26, these descriptors are formed by a key-value pair, respectively the *@schemeIdUri* and the *@value* attribute. *@schemeIdUri* must contain a URI that specify the syntax and the semantics expressed within the *@value* attribute; *@value* contains a formatted string following the specification of the given URI.

In case of SRD, the *@schemeIdUri* is *urn:mpeg:dash:srd:2014* and the *@value* is formatted as a Comma-Separated Values (CSV) list containing the following parameters:

1. THE IMMERSIVE STREAMING TECHNOLOGY

- *source_id*, defines the system coordinate in use;
- *object_x*, states the x-axis coordinate of the top-left corner for the associated tile;
- *object_y*, states the y-axis coordinate of the top-left corner for the associated tile;
- *object_width*, specifies the width of the associated tile;
- *object_height*, specifies the height of the associated tile;
- *total_width*, specifies the maximum extent for the system coordinate along the x-axis. The summation of *object_x* and *object_width* must be less than this value for each tile;
- *total_height*, specifies the maximum extent for the system coordinate along the y-axis. The summation of *object_y* and *object_height* must be less than this value for each tile;
- *spatial_set_id*, identifies the multimedia content set.

Each of this parameter is represented as a decimal non-negative integer. The first five parameters (i.e. *source_id*, *object_x*, *object_y*, *object_width*, and *object_height*) are mandatory in the *@value* field of each descriptor, while *total_width* and *total_height* must be present in at least one of the descriptors associated with a given *source_id* *spatial_set_id* is optional.

Moreover, the SRD specification contemplates different use cases associated with the *Essential Property* and *Supplemental Property* elements. Specifically, DASH clients are allowed to discard the multimedia content associated with the *Essential Property* element if *@schemeIdUri* is not correctly recognized. On the contrary, DASH clients can further process multimedia contents associated with the *Supplemental Property*, even in the case of the *@schemeIdUri* is not correctly handled. In this way, content creators are enabled to deliver multimedia contents even if SRD is not fully supported by client without disrupting the streaming service. This feature is particularly useful in the case of streaming service has to be guaranteed to both legacy and SRD-aware DASH clients.

1.5.2.2 MPEG HEVC Motion Constrained Tile Set

As thoroughly discussed in Section [1.3.4](#), HEVC was designed to provide advanced encoding features for higher parallelism and better resulting compression. In particular, the *Tiles* feature allows the frames composing a video to be spatially divided

1.6 A standard for Virtual Reality: MPEG Omnidirectional Media Format

into rectangular regions, where both intra-picture and entropy decoding prediction dependencies across tile boundaries within the same picture are constrained [66]. This enables each tile in the reference picture to be decoded in independent fashion. However, inter-prediction dependencies are not constrained with respect to tile boundaries, i.e. a particular tile in a non-reference picture could require data belonging to different tiles in the reference picture. This can hinder the implementation of tiled streaming approaches based on RoIs [133]. For the purposes to meet the needs of emerging RoI-based techniques, the MPEG video experts proposed an amendment [134] to the original HEVC standard, introducing the concept of Motion Constrained Tile Set (MCTS).

A MCTS is a set of tiles, in the reference and in the subsequent pictures, for which the inter-prediction process can be performed only with picture data belonging to the set of tiles itself [134], i.e. the inter-prediction is disabled across the boundaries of the specific set of tiles. In particular, the MCTS amendment to the HEVC standard [134] provides the syntax element to enable the signalling the presence of one or more MCTSs directly into the bitstream.

In this way a HEVC decoder can correctly decode a specified MCTS without the need of decoding the entire video content [133]. In simple words, a MCTS represents a fully independent decodable spatial partitioning, which exactly matches with the definition given in [1.5.2]. Indeed, this is an evaluable feature commonly exploited by advanced viewport-dependent techniques [135] [136] [137] [138] [139],

1.6 A standard for Virtual Reality: MPEG Omnidirectional Media Format

As stated in Section [1.5], the industrial interest on emerging VR and AR applications is steadily growing. This increased interest has led several non-interoperable VR platform to be developed and deployed. This aspect is particularly noticeable into the research field of projection formats, where different solutions have been proposed and used, as discussed in [1.5.1]. In a effort for solving this issue, the MPEG has recently started the development of a standard specifically designed for VR applications. This standard, called OMAF [140] is the first standard specifically designed to enable Immersive video applications over the web. To this end, OMAF extends well-established international standards such as DASH and ISO/BMFF. Even if OMAF is able to manage

1. THE IMMERSIVE STREAMING TECHNOLOGY

different kind of media - video, audio, and timed text are supported herein - the main contribution was on the definition of a standardized format to deliver OV contents.

The actual revision of OMAF standard [140] supports Immersive applications with 3-DoF - *yaw*, *pitch* and *roll* orientation - but is planned to provide support for 6-DoF within future versions. Going into detail, OMAF assumes that the OV is textured on the inside surface of a sphere with the viewer collocated at the center. Multiple media contents which have to be presented to the user at the same time share a right-handed global coordinate system and an initial viewing orientation. Additionally, each media can define specific local coordinates. The *yaw*, *pitch*, and *roll* rotational coordinate - stored in dedicated OMAF *metadata* - are used to define the position of the local coordinate system with respect to the global coordinate system. The design choice to use both global and local independent coordinate systems could be useful for facilitating the stitching operation between tiles, thus improving the resulting perceived picture quality.

Furthermore, OMAF allows signalling information about the projection format used to map OV contents. The first draft of OMAF [140] defines the support for the most used OV formats such as ERP, CMP and Fish Eye. Advanced video manipulation techniques such as *tiled streaming* are supported through the Region Wise Packing (RWP) feature. Such an example, stereoscopic OVs - in the form of side-by-side, top-bottom or temporally interleaved frame - are also supported thanks to the RWP feature. Generally, Region Wise Packing (RWP) enables the transmission of additional metadata concerning the optional manipulation of a OV content after projection. Such metadata contains information about the position and size of different tiles in both projected and packed pictures, jointly with indications of the possible rotation and mirroring applied. Additionally, a particularly useful feature provided by OMAF is the Region Wise Quality Ranking (RWQR), through which it is possible to indicate the visual quality relative to each tile.

OMAF defines mechanism to pack the aforementioned metadata into the common DASH and ISOBMFF standard. The following paragraph summarizes the modifications introduced by OMAF to the MPEG standards.

1.6 A standard for Virtual Reality: MPEG Omnidirectional Media Format

Descriptor name	Description
<i>FramePacking element</i>	Frame packing format for Stereoscopic OV content
<i>PF descriptor</i>	Indicates projection format(s) in use for that particular OV
<i>RWPK descriptor</i>	Indicates whether RWP has been applied
<i>CC descriptor</i>	Content coverage
<i>SRQR descriptor</i>	RWQR information for sphere regions
<i>2DQR descriptor</i>	RWQR information for rectangular regions on decoded frames
<i>FOMV descriptor</i>	Indicates fisheye omnidirectional video. Additionally, common twolens setups (monoscopic 360, stereoscopic 180) can be indicated.

Table 1.1: Additional MPD descriptors defined in OMAF

1.6.0.1 OMAF ISOBMFF and DASH extension

As anticipated in the previous section, OMAF standard aims at enabling the streaming for OV contents over the internet. Streaming OVs require, on the one hand, mechanisms enabling the selection of OV variant and, on the other hand, procedures needed for the correct rendering of OV content on user screens. To this end, OMAF defines extensions for both DASH and ISOBMFF MPEG standards. In the same way as with SRD, OMAF specifies additional DASH *descriptors* to carry out information about specific OV. Table [1.1](#) lists the descriptors defined for DASH [\[141\]](#)

Furthermore, enabling correct playing and rendering demands that information about the OV being specified also at video container level. The ISOBMFF standard defines the *restricted video sample entry* ('*resv*') type specifically for such video tracks requiring post-processing operations after decoding, with one or more *scheme types* specifying the required post-processing operations. OMAF extends the ISOBMFF *resv* by defining additional *scheme types* that allows the management of OV contents. The *scheme types* defined by OMAF are listed into Table [1.2](#).

In this way, information such as used projection format, rotation, mirroring, tile

1. THE IMMERSIVE STREAMING TECHNOLOGY

Scheme Type	Description
<i>podv</i>	Generic OV content
<i>erpv</i>	Type scheme supporting simple ERP OV contents with single RWP packed region
<i>ercm</i>	Type scheme supporting ERP and CMP OV contents, multiple RWP packed regions allowed
<i>fodv</i>	Type scheme supporting <i>fisheye</i> OV

Table 1.2: Additional *Restricted Scheme Types* defined in OMAF

position and size are easily integrated into the ISOBMFF container format as metadata.

Moreover, OMAF introduces the concept of *Video Profiles* aiming at supporting the streaming of OV contents. In particular, OMAF defines the following Video Profiles:

- *Viewport independent 360* streaming: OV contents are streamed with uniform visual quality;
- *Viewport dependent 360* streaming: the visual quality shown by the viewport is higher with respect to other regions of the panorama. In particular, OMAF supports the following methods used for implementing the viewport dependent streaming:
 - Viewport-specific 360 streaming: multiple DASH *Representations* of the same OV are encoded, each one with a viewing direction pointing to a RoI having higher quality. The VR player selects the best representation according both actual viewport and RoI;
 - Tile-based viewport-dependent 360 streaming: the OV content is partitioned into several tiles. The server can bundle tiles in several DASH *Representations* to reduce the number of HTTP GET required. The VR player selects the set of tiles that covers the actual viewport.

Table [L.3](#) shows the codec settings supported by OMAF Profiles.

On the one hand, *Viewport independent* streaming allows to reuse existing VR player implementations with little modification. On the other hand, *viewport dependent* enables advanced OV manipulation techniques.

1.6 A standard for Virtual Reality: MPEG Omnidirectional Media Format

Codec	OMAF video Profile	Codec Profile	Bit depth	Decoding capacity	Scheme types
HEVC	<i>viewport-independent</i>	Main 10	$\leq 10bits$	4K@60Hz	<i>erpv</i>
	<i>viewport-dependent</i>	Main 10	$\leq 10bits$	4K@60Hz	<i>erpv/ercm</i>
AVC	<i>viewport-dependent</i>	Progressive High	8 bits	4K@30Hz	<i>erpv/ercm</i>

Table 1.3: OMAF Video Profiles

In an attempt to support both newer and legacy devices and implementations, OMAF allows the encoding of OVs with both HEVC and AVC video codec. Nevertheless, it is worth to remark that common AVC decoder implementations ask for a separate decoding instance each tile, while HEVC allows the decoding of multiple tiles with a single instance.

Indeed, as introduced in the paragraph [1.5.2.2](#), HEVC supports natively *tiled streaming* through the MCTS feature. In summary, HEVC MCTS enables partitioning of a picture into a regular grid of independently decodable tiles set by restricting encoding operations within the same tile set in the current and the reference picture. The result is that tiles can be removed from the bitstream without breaking decoding. In AVC, a similar technique can be achieved with the usage of vertical *slices* - requiring extra effort in order to restrict the inter-prediction process within each slice - or by encoding each tile in a separate track.

In OMAF tile-based viewport-dependent streaming, each tile is encoded as MCTS, stored on server as HEVC-compliant sub-picture track and listed into the MPD as a single *Representation*. The choice to use HEVC sub-picture tracks - marked with the sample entry type *'hvc1'* - instead of HEVC tile tracks - sample entry type *'hvt1'* - eases decoding with multiple instances. In this way, OV contents can be encoded with HEVC and AVC and streamed with a single DASH MPD.

Nevertheless, single-instance decoding can be enabled through the streaming of an additional *extractor track*, as defined in the second amendment of the ISO/BMFF encapsulation format [\[142\]](#). An extractor track is formed by repeating the PS and slice

header information contained in other video tracks jointly with a byte range referring to coded video data (MCTSs or *slices*). Therefore, players are able to merge coded video data into different tracks in a valid HEVC or AVC bitstream. OMAF recommends the use on an extractor track for each distinct viewing orientation [141].

1.7 Control systems for adaptive video streaming

This thesis considers control systems adopting the stream-switching (as known as MBR or ABR) approach. As anticipated in section 1.3, ABR is the dominant technology today and is used by all Internet video distribution platforms. Such systems require the server to encode the same video content into different bitrate levels (or representations), thus forming a discrete set $L = l_0, l_1, \dots, l_{N-1}, (l_i < l_{i+1})$, the set of video layers. Each video layer is then logically, or physically, divided into segments of constant duration (typically on the order of seconds). Main task of the control algorithm is to determine the video level to be requested at each segment download. The ultimate goal of the control strategy is to maximize the QoE perceived by viewers.

It has been shown by several independent studies that in order to improve viewers perceived QoE, it is necessary to pursue the following objectives (in descending order of importance) [143] [48]:

1. avoid interruptions in playback (rebuffering events);
2. maximize video quality (level or bitrate);
3. minimize start-up time;
4. minimize the number of video level switches.

To achieve these requirements, the conventional approach is to jointly use two algorithms:

1. an algorithm for the dynamic selection of the video level, which should ideally match the available bandwidth;
2. a playout buffer controller that is used to absorb bandwidth variations and avoid interruptions in playback.

The playout buffer is the portion of storage dedicated to storing the video segments that will be played by the player. The amount of video stored in the buffer expressed in seconds is called the playout buffer level and its instantaneous value is denoted by

1.7 Control systems for adaptive video streaming

the symbol $q(t)$. In general, based on the fluid-flow model described in [45], the buffer level can be modeled as described in the following differential equation:

$$dq(t) = f_r(t) - d_r(t), \quad (1.16)$$

where $f_r(t)$ is the buffer *filling rate*, i.e. the frequency with which new segments are added, while $d_r(t)$ is the *draining rate*, i.e. the speed with which video segments are played by the player. If a video segment of duration dt_v is downloaded in time dt and stored in the buffer, the filling rate is $f_r(t) = \frac{dt_v}{dt}$. Given dD the length of the video segment expressed in bytes, it is possible to express $f_r(t)$ as

$$f_r(t) = \frac{r(t)}{l(t)}, \quad (1.17)$$

being $r(t) = \frac{dD}{dt}$ the download rate experienced by the player, and $l(t) = \frac{dD}{dt_v}$ the video level (expressed as bitrate) of the downloaded video segment decided by the controller. The playout buffer is emptied by the player during playback: if t seconds of video are played in time t , $d_r(t)$ is worth 1 when playing, 0 when paused.

It is therefore clear that the appropriate design of a playout buffer control algorithm heavily affects the reduction of the rebuffering events probability, hence the overall user satisfaction.

Classical playout buffer control algorithms are designed by taking one of two different approaches:

- Rate-based approach, in which the buffer is controlled based on the received rate;
- Level-based approach, in which the buffer is controlled based on the received video level.

In the next paragraphs a review of the traditional methodologies used for challenging the playout control problem in adaptive streaming along with an analysis of the particularities of playout buffer control in the case of immersive content streaming will be presented.

1.7.1 Rate-based approaches

In the rate-based approaches, control over the playout buffer is achieved by choosing the video level $l(t)$ as the highest level $l \in L$ lower than the received rate r . These

1. THE IMMERSIVE STREAMING TECHNOLOGY

systems consider the end-to-end bandwidth $r(t)$ constant during playback and equal to r . Considering that the playout buffer fill rate equals to [1.17](#), in this case it would be greater than 1. Thus, this approaches lead to a queue always growing, thus wasting resources or, in the worst case, sending the buffer into overflow, with a destructive effect on the system. For these reasons $r(t)$ must be on average imposed equal to $l(t)$ to keep $q(t)$ at a predefined set value q_T . To this end, idle periods between the download of two consecutive video segments have to be inserted to ensure that the received rate $r(t)$ to be equal to $l(t)$. In other words, the client alternates between the ON and OFF phases: during the ON period, the client receives at a rate $r(t) = r$, while during the OFF period it remains idle, i.e., $r(t) = 0$. In this way, the average rate received in an ON-OFF period can be made equal to the level of the selected video $l(t)$ by correctly setting the OFF duration. The advantage of this approach is that if the end-to-end bandwidth is constant, the video level is kept constant and the queue keeps track of the set point.

Despite its simplicity, this approach has two major drawbacks that have been widely studied in the literature:

- the available bandwidth is always underutilized;
- it has been experimentally shown that the ON-OFF traffic model heavily penalizes bandwidth utilization in the case of concurrent video streams [\[144\]](#), [\[145\]](#) [\[146\]](#).

The first problem can significantly degrade the perceived QoE in case of the distance between layers is high. The second problem, known in the literature as the downward spiral effect [\[145\]](#), [\[48\]](#), can lead to an even worse effect on QoE.

1.7.2 Level-based approaches

In the level-based approaches, the ON-OFF traffic pattern is eliminated by downloading video segments sequentially, keeping the download rate $r(t)$ constant and always equal to the bandwidth B . This ensures full utilization and fair sharing of the available bandwidth. In this way, full utilization and fairness in sharing the available bandwidth is guaranteed in the case of competing video streams, thus eliminating the downward spiral effect.

Control over the length $q(t)$ of the playout buffer is performed by varying the video level $l(t)$ in such a way as to keep it within a discrete range $[q_L, q_H]$. In fact, the

quantized nature of $l(t)$ does not allow for continuous adjustment of $q(t)$. The major drawback of this approach is that, in the absence of appropriate precautions, at steady state fluctuations in the video level (and thus also in the reproduced quality) occur even with a constant bandwidth.

In general, each of these approaches performs well under certain conditions but not under others. In particular, rate-based approaches are best at startup and when the end-to-end bandwidth is stable, while buffer-based approaches are more robust at steady state and in the presence of high variability in the available bandwidth. For these reasons, the academic community has produced control algorithms using a hybrid approach. Among the various proposals, Model Predictive Control (MPC) algorithms [52] characterize adaptive streaming systems using stochastic optimal control methods. This class of algorithms is based on maximizing an objective function that uses both throughput estimates and buffer occupancy information as input variables to select the video level to be requested. The estimates can be computed either dynamically (over a longer or shorter range of video segments), or based on tabulated values. Given the higher accuracy of the estimates of bandwidth and control over the buffer playout, they are designed to be implemented client-side, which makes this type of algorithm adhere to the DASH standard. However, errors in throughput (predictor) estimation severely affect their performance. This is particularly evident in the case of wireless networks [52].

Another approach, proposed in [147], overcomes the problem of throughput estimation accuracy by using modern reinforcement learning techniques to learn the most appropriate bitrate control policy dynamically without the need for prior knowledge of the transport system. Although the results obtained in [147] are very interesting in terms of QoE, pure reinforcement learning techniques may present robustness issues in case of the system is faced with scenarios that are very dissimilar to the scenarios explored in the training phase.

From an architectural point of view, the control algorithms adhering to the DASH standard are client-driven. This can lead to a sub-optimal use of network resources in the case of concurrent video streams [148]. In particular, client-driven algorithms share network resources equally among all involved streams. Thus, the differential QoE needs of each or particular groups of streams cannot be taken into account. For these reasons it is necessary an interaction between video client and video streaming provider.

1. THE IMMERSIVE STREAMING TECHNOLOGY

Network-assisted systems try to overcome these problems by including devices in the network that can collect information from different clients and provide global Video Control Plane (VCP) over the entire video streaming system [149].

1.7.3 Control problem on Immersive Video Streaming

Compared to control systems for 2D video streaming, immersive streaming systems have unique peculiarities that differentiate them from conventional streaming systems. First, immersive content streaming systems require higher resolution video and therefore a higher encoding bitrate to ensure quality levels comparable to 2D video. Moreover, the user views only a portion of the transmitted video (known as viewport), which opens the door to new optimization strategies based on the concept of *view*. A view represents a particular version of the video in which the portion of the video - the so-called RoI - that is supposed to constitute the viewport is encoded at a higher quality, while the remaining part is encoded at a lower quality.

For these reasons, control systems for the streaming of immersive contents need of ad hoc algorithms for selecting the best view to download among those available in order to guarantee to the user the highest possible visual quality. It is worth noting that the problem of selecting the best view can be formulated as a prediction problem. In general, the scientific literature has identified two possible mechanisms of view selection [40]:

- Content-agnostic algorithms: these approaches are based on the analysis of historical data movements of the viewers for predicting future viewing positions;
- Content-aware algorithms: these approaches are based on the analysis of the recorded 360 scene.

Both types of algorithms have advantages and disadvantages. On the one hand, content-agnostic techniques use well-established algorithms for pursuing prediction, such as averaging [150], linear regression [150] [151], advanced localization techniques [152], machine learning algorithms based on clustering [153] [154] [135] [155] [156] [157] and attention-based neural encoder-decoder networks [156] [158]. At the moment, the prediction accuracy of these approaches is poor for prediction horizon longer than 3.5s [156].

On the other hand, content-aware methods - such as, the one utilizing saliency maps - are considered as a key technology for enabling Immersive streaming applications in

1.7 Control systems for adaptive video streaming

daily life, because are able to achieve greater performance on prediction accuracy. In general, these methods combine saliency patterns and user gaze information in the 360 scene to perform the prediction. For instance, the authors in [159] designed PanoSal-Net, a framework which uses information from both HMD sensor and saliency maps for predicting the user fixation in 360 video contents. Moreover, in [160] the authors proposed an approach for predicting the head movement of viewers based on deep reinforcement learning techniques. The conceived framework was composed by an offline deep reinforcement learning model to extract saliency information from multiple head movement traces, and an online module, performing the head movement prediction for the specific user. In [161], the authors proposed to use the history scan path and the image features for performing the gaze prediction. In summary, they collected a dataset of head movement traces exploiting the eye-tracking capabilities offered by the HTC VIVE headset. The dataset has been used for computing saliency maps at three spatial scales: the actual gaze direction, the viewport and the entire image. The saliency maps and the images have been fed to a Convolutional Neural Network for extracting image features, while Long-Short Term Memory (LSTM) neural network has been used for estimating the watching pattern of the user. Other works [162] [163] [164] explore the use of motion maps for gaze prediction. Nevertheless, the use of motion maps need further investigations for since possibly different motion patterns lower the achieved accuracy.

Anyways, a systematic performance comparison between the two mechanisms is not available in the literature at the moment. Furthermore, the interaction between view selection and bitrate selection algorithms is an open research topic. In particular, a control system for immersive content must consider how the two algorithms should cooperate in order to obtain the best trade-off between QoE perceived by the user and resources used for the delivery of the immersive streaming service.

1. THE IMMERSIVE STREAMING TECHNOLOGY

2

Reducing the Network Bandwidth Requirements for 360 Immersive Video Streaming

In this chapter, a scaling technique to reduce bandwidth requirements to stream omnidirectional videos is presented. An experimental investigation of the proposed approach has shown that it is possible to obtain a reduction of the required bitrate up to around 50% while gracefully degrading visual quality far from the RoI. The scientific results object of the following chapter have been disseminated in two scientific works [\[165\]](#) [\[131\]](#).

2.1 Background

Recently, different approaches have been proposed to reduce the required network bandwidth to stream the content which are summarized in the following. One popular design strategy employed today is to stream to the user only a portion of the video, the one falling in the current user's FoV, i.e., the RoI. One way to implement this approach is using the *slicing* technique which divides the video into several portions which are encoded and stored separately in different bitstreams. The advantage of this approach stems from its implementation simplicity. The drawback is that a RoI may span multiple slices, each one requiring one decoding process running on the client device. Consequently, this solution cannot be easily implemented in mobile devices. Moreover, the

2. REDUCING THE NETWORK BANDWIDTH REQUIREMENTS FOR 360 IMMERSIVE VIDEO STREAMING

client has to download in parallel the slices composing the RoI, making the adaptive streaming algorithm considerably more complex.

A new approach not requiring separate decoding process is *tiling*, a concept which has been introduced in HEVC and recently also considered for DASH-compliant video delivery systems [166, 167]. This approach requires the video to be spatially divided into several *tiles* which are encoded independently and possibly stored into a single bitstream. Spatial relationship between different tiles can be embedded into bistream using either i) the OMAF [168] extension or ii) integrated into the MPD employing SRD. Either way, the client can decide to request a subset of the available tiles (the ones composing the RoI) and a single process is able to decode the received compressed bitstream. However, tiling does not allow varying the resolution of the representations, but only their bitrate. As a consequence, the resolution of each tile must remain constant, i.e. the tile grid cannot change across representations [166]. Another limitation is that tiling efficiency decreases when increasing the number of tiles [129]. Most importantly, when sudden changes of the viewpoint occur, video segments of new tiles should be quickly downloaded and rebuffering events might occur in the case those segments are not downloaded in time. At the time of this writing, encoders supporting tiling are still at the experimental stage, in particular with respect to hardware encoding [169]. Indeed, the FFmpeg multimedia framework has only recently introduced the HEVC tiling feature in the VA-API hardware encoding library [170], while NVIDIA still lacks supporting the HEVC tiling in their Software Development Kit (SDK) [171]. Finally, the open source *kvazaar* HEVC encoder, developed by the leading academic video coding group Ultra Video Group [172], has advanced tiling support but it lacks in hardware encoding, thus hindering its wide adoption in the mobile ecosystem.

A different class of approaches leverages a *scaling technique* to reduce the bitrate to encode the projected 360 video [173, 174]. In [173], the authors propose a technique exploiting the downscaling operation to realize a specific mixed-resolution packing for 360 video streaming. The bitrate reduction here is achieved by varying the used Quantization Parameter to encode each tile, while the downscaling operation is exploited to rearrange some portion of the 360 video properly. In [174], a gaussian pyramid projection mapping technique is conceived to provide viewport-adaptivity. Once the RoI is identified, the gaussian pyramid is implemented by halving the resolution recursively in the areas around the RoI.

Starting from these research premises, in the following a scaling technique to reduce bandwidth requirements to stream omni-directional videos is presented, having the following main features:

1. it achieves bitrate reductions by aggressively reducing the horizontal resolution of the areas outside the main RoI;
2. it employs an approach that is encoder-agnostic;
3. it can be easily adopted using standard technologies already available in the vast majority of mobile platforms and devices.

Differently from the approach proposed in [174], in this research work the bitrate reduction is obtained by applying downscaling homogeneously to the regions outside the RoI. Such an approach makes the implementation considerably simpler, a particularly important aspect to empower live streaming of 360 videos.

2.2 Proposed Approach

The de-facto standard employed today in the industry is the *MPEG* DASH protocol which allows clients to dynamically adapt the video bitrate to the time-varying network bandwidth. The video content is stored on a standard HTTP server and a client fetches the video by employing an HTTP connection. The video content is encoded at different bitrate *levels* (or *representations*) which form the *video levels set* $\mathcal{L} = \{l_1, l_2, \dots, l_M\}$ with $l_i < l_{i+1}$ [47]. At the client, a control algorithm dynamically selects the video level to be streamed at each segment download.

The content generation in the case of omni-directional videos differs significantly from the one employed for classical 2D videos. In particular, 360° cameras capture a spherical scene (the omni-directional video) that need to be projected onto the 2D plane (the projected video) in order to be encoded. It is worth to mention that only a small portion (roughly one-sixth of the video resolution) of the projected video falls in the users' viewport, i.e. the part of the video which is currently visualized by the user. To make a concrete example, in order to deliver a video content with a viewport resolution of 1080p, the video resolution of the projected video has to be larger than 6480p that is a resolution larger than 8K ultra HD. Indeed, the encoding of such a large resolution video at high quality might result in a too large video bitrate. Consequently,

2. REDUCING THE NETWORK BANDWIDTH REQUIREMENTS FOR 360 IMMERSIVE VIDEO STREAMING

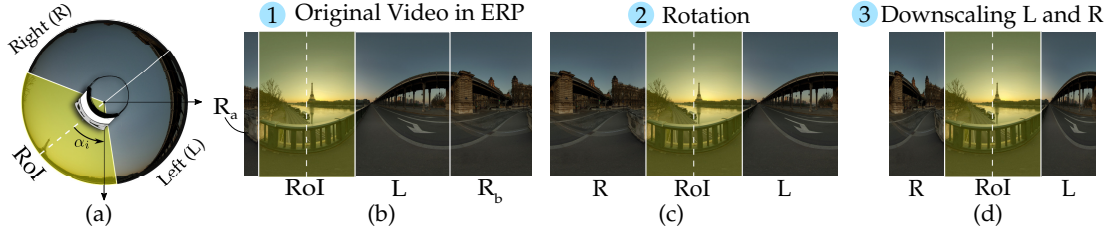


Figure 2.1: Approach to generate the i -th RoI representation

streaming the encoded projected video at full resolution entails a remarkable waste of network bandwidth.

The idea to reduce the video bitrate is that only the parts of the scene that are considered to be a RoI should be encoded at a high quality. This work considers the RoI as the portion of the video falling in the current user’s viewport at a given time. As such, the RoI changes over time and depends on the scene and on the user’s behavior during the video playback.

In a nutshell, the idea is to generate from one projected video a number N of versions, the *views*, that encode RoIs at different positions. All the views constitutes the *views set* $\mathcal{V} = \{v_1, \dots, v_N\}$. Now each view $v_i \in \mathcal{V}$ is encoded into M video representations at different bitrates l_j (and resolutions) constituting the video *level set* $\mathcal{L} = \{l_1, \dots, l_M\}$. At the end of this procedure, the DASH Server stores and indexes a set of representations $\mathcal{R} = \mathcal{V} \times \mathcal{L}$ composed of $N \cdot M$ files. In the following, details of the methodology proposed to produce the views v_i will be provided.

Without loss of generality, the original uncompressed scene has been considered being produced in ERP format, which is by far the most popular output format for 360° cameras. Notice that this is not a limitation since any other format is in principle supported by using format adapters filters.¹ In order to produce the different views v_i it is required to first manipulate the original ERP video. To the purpose, the RoI has been identified as the *spherical lune* (a slice of the sphere) with a dihedral angle (i.e., the FoV angular width) equal to 120°, centered at a particular yaw angle α_i . Let us consider Figure 2.1a that shows a user seen from above (the user is considered to be in the center of the sphere) with his head turned left so that his FoV is centered at a certain yaw angle α_i which falls into a specific RoI (the shaded area). The regions

¹<https://trac.ffmpeg.org/wiki/RemapFilter>

outside the RoI, namely the ones at its left (L) and its right (R), are divided into two spherical lunes of equal dihedral angle. Since the video is represented in ERP format, each spherical lune maps to a particular vertical strip of the video as shown in Figure 2.1b. The video in ERP format is manipulated in such a way that the RoI is always placed in the center of the frame as Figure 2.1c shows. The idea is to downscale the portions of the video outside the RoI, which are less likely to be in the user’s FoV, to reduce the required encoding bitrate as shown in Figure 2.1d.

The choices made for the design of our strategy are motivated in the following. First, the RoI has been identified as the spherical lune because more complex strategies (such as ones employing spherical sectors [174, 175]) may introduce inefficiencies into the intra-frame operations, leading to higher bitrate requirements [166, 176]. Moreover, maintaining the RoI at the center of the frame—applying a rotation before downscaling—allows to better exploit the motion compensation algorithm by keeping the continuity between the scaled and non-scaled areas [166, 176]. Finally, the usage of the *downscaling* operation—instead of *HEVC tiling*—is due to the fact that this technique is 1) independent of the employed codec, 2) can be efficiently handled by hardware decoders at the client-side, 3) can use well-established algorithms (interpolation, filtering, etc.) to improve the resulting video quality.

2.3 Methodology

The content generation mechanism proposed in Section 2.2 has been implemented using a filter chain using FFMPEG¹. A video catalog composed of ten benchmark videos having a 4K resolution (i.e., 3840×2048 , 30 fps) has been produced. The videos were selected to produce a catalog sufficiently representative of different video categories and features. To investigate the relationship between the obtainable bitrate reduction and the resulting video quality, each video in the catalog has been encoded using the reference FFMPEG H.264 encoder (*libx264*)² with a *Constant Rate Factor (CRF)* parameter equal to 20 before applying the downscaling algorithm.

As described in Section 2.2, for each view $v_i \in \mathcal{V}$ the regions outside the RoI are downscaled in order to reduce the encoding bitrate. The *downscale factor* d is defined

¹<https://ffmpeg.org/ffmpeg-filters.html>

²<https://trac.ffmpeg.org/wiki/Encode/H.264>

2. REDUCING THE NETWORK BANDWIDTH REQUIREMENTS FOR 360 IMMERSIVE VIDEO STREAMING

as the ratio between the width of the downsampled video and the original video width w , i.e. $d = (2w_d + w_{\text{RoI}})/w$, where w_d is the width of the downsampled regions outside the RoI and w_{RoI} is the width of the RoI. Since the catalog is composed of video with a resolution equal to 3840×2048 , the resolution of each of the three vertical strips in which the video is divided (left, RoI, right) is equal to 1280×2048 . The downsampled width w_d varies in the set $\{240\text{px}, 480\text{px}, 720\text{px}, 1080\text{px}\}$. The GOP parameter has been set equal to 60 frames, a typical setting commonly used by video streaming services. In order to quantitatively assess the video quality between the manipulated video (upscaled to the original resolution) and the original video in the benchmark video catalog, the PSNR and SSIM FFMPEG filters have been employed.

2.4 Considered scenarios

The experimental evaluation has been carried out into two scenarios:

1. the *CRF scenario*, in which the encoding parameters have been set to constant video quality, aiming at showing the impact of the conceived mechanism on the overall video quality;
2. the *Average Bitrate (ABR) scenario*, in which the encoding parameters have been set to produce a constant average bitrate, to explore the relationship between the proposed mechanism and video quality with a constraint on average video bitrate.

The workflow of the CRF scenario is described in details in Algorithm [1](#). In a nutshell, for each video, view, and considered downscale factor, the procedure described in Section [2.2](#) is carried out to produce downsampled video versions. For each downsampled video, the visual quality has been estimated by using the well-established PSNR and SSIM metrics, while the average bitrate has been measured in bit/s. At the end, for each downscale factor d the average bitrate reduction factor has been derived as $\hat{r}_d = \mathbb{E}[r_d^{(i)}]$.

In the ABR scenario, the average bitrate reduction factors \hat{r}_d have been employed to set for each view i a target average bitrate $\bar{b}_d^{(i)}$ equal to $(1 - \hat{r}_d)b_o^{(i)}$. The same workflow described in Algorithm [1](#) has been used but, instead of encoding the video using a CRF (line 8), the encoder has been set in ABR mode with a target bitrate $\bar{b}_d^{(i)}$. Such an approach has been considered because DASH systems typically produce video content by encoding videos in ABR mode to limit bitrate fluctuations.

Algorithm 1: Pseudo-code of CRF scenario

```

1 for each video do
2   for  $i=0$  to  $N$  do
3     Generate  $i$ -th view  $v_i$ ;
4     Transcode the  $i$ -th view with CRF=20;
5     Measure the average bitrate  $b_o^{(i)}$ ;
6     for each downscale factor  $d$  do
7       Downscale view  $v_i$  at factor  $d$ ;
8       Encode downsampled video with CRF=20;
9       Measure the average bitrate  $b_d^{(i)}$  and compute the bitrate reduction
        factor  $r_d^{(i)} = b_d^{(i)} / b_o^{(i)}$ ;
10      Upscale to the original resolution and measure  $SSIM_d^{(i)}$  function of
        viewport yaw angle;
11    end
12  end
13 end

```

2.5 Results

In the CRF scenario, the estimated video quality at different viewport yaw angles has been investigated for each considered downscale factor d . In CRF mode, the encoder is free to vary the output bitrate in order to reach a given video quality. As mentioned in the previous section, in this scenario a CRF equal to 20 has been considered.

Figures 2.2 and 2.3 shows the estimated video quality averaged over the considered videos in function of yaw angles. In Figure 2.2 is expressed in the PSNR visual quality metric, while Figure 2.3 reports the SSIM visual quality metric.

The Figures 2.2a and 2.3a clearly shows that video quality is maximal at the center of the viewport (yaw angle equal to 180) and it gracefully decreases when the angular distance from the center increases. Moreover, as expected, the slope of the video quality curves gets steeper when the scaling factor increases (i.e., when the downsampled resolution decreases). Thus, when the scaling factor is higher, the video quality degrades faster when the user moves away from the center of the RoI. Table 2.1 reports the average bitrate reduction \hat{r}_d (expressed in percentage) measured for each of the considered downscale factors. The results show that the proposed approach provides a

2. REDUCING THE NETWORK BANDWIDTH REQUIREMENTS FOR 360 IMMERSIVE VIDEO STREAMING

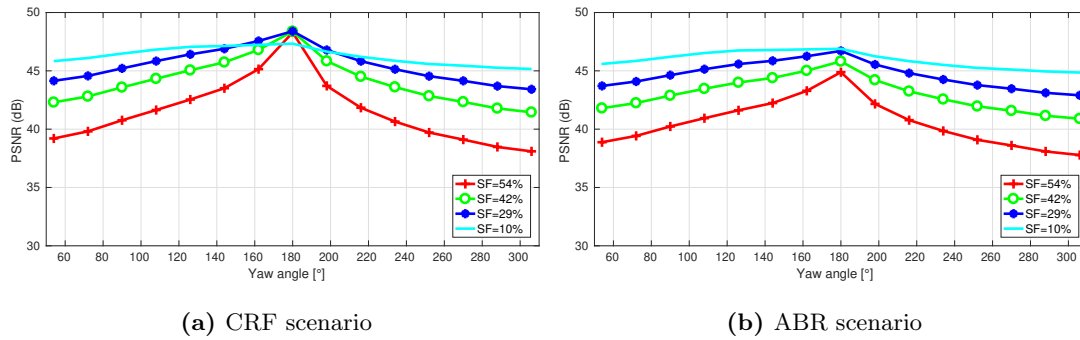


Figure 2.2: Average PSNR function of the viewport yaw angle

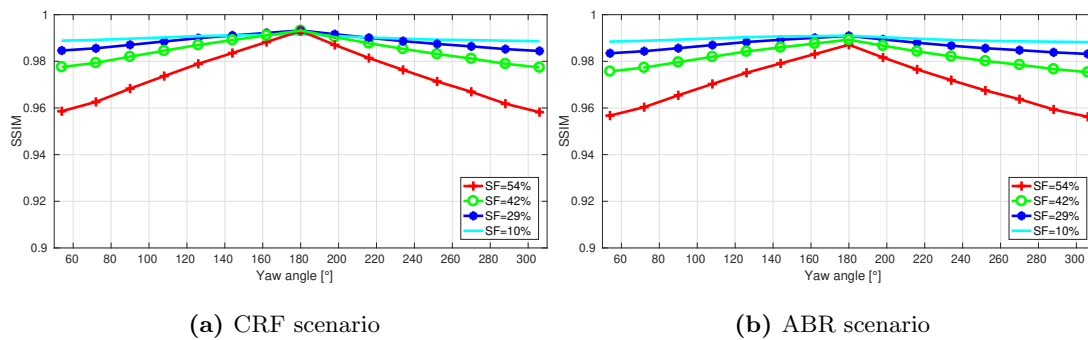


Figure 2.3: Average SSIM function of the viewport yaw angle

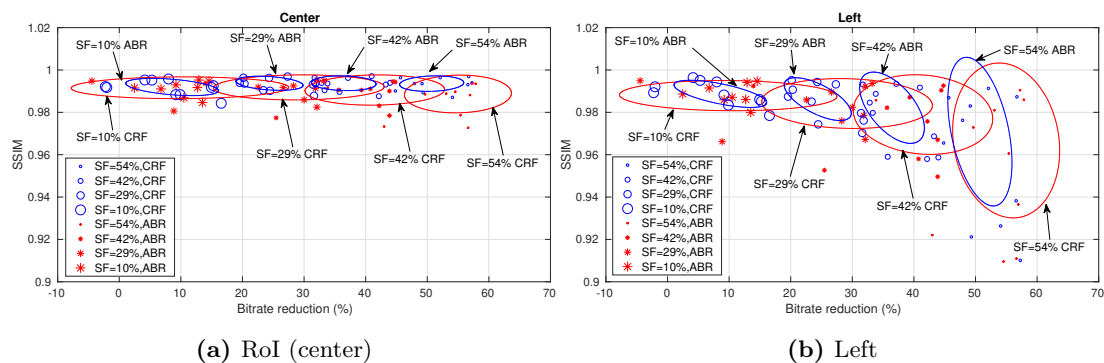


Figure 2.4: SSIM as a function of the Bitrate reduction percentage

Downscale factor d (%)	Downscaled resolution (px)	Average Bitrate reduction (%)
54.17	240 px	51.3 (48.0-54.7)
41.67	480 px	37.49 (34.1-40.8)
29.17	720 px	25.44 (21.9-28.9)
10.42	1080 px	7.90 (3.0-12.7)

Table 2.1: Average bitrate reduction (with 95% confidence interval reported in the parentheses) for the considered downscaled resolutions in the case of CRF=20.

percentage bitrate reduction scaling almost linearly with the downscale factor d . Notice also that confidence intervals are quite tight, indicating that the proposed scheme is not content-sensitive.

In the ABR scenario, the impact on the video quality of the proposed bitrate reduction mechanism has been evaluated when a bitrate constraint is added as described in the previous section. Figures 2.2b and 2.3b shows that driving the encoder in ABR mode produces a smooth quality transition between lateral and RoI regions, gracefully degrading the video quality. Compared to the previous scenario, the video is only slightly affected. In particular, it has been found a maximum video quality loss in term of PSNR and SSIM respectively of around 4dB and 0.005. This result indicates that the proposed content generation scheme performs satisfactorily also when the encoder is driven using a target average bitrate.

We conclude this section by comparing the obtained results in the two considered scenarios. Figures 2.4a and 2.4b show scatter plots of the obtained SSIM against the measured bitrate reduction respectively when the yaw angle is at the center of the RoI or at the left¹. Scatter plots for PSNR metric show similar results, thus we decide to omit even though same qualitative insights can be drawn by analyzing the SSIM.

In particular, each data point represents the obtained (bitrate reduction, SSIM) for one video of the catalog encoded at a particular scaling factor and encoding strategy (CRF (\circ marker) or ABR ($*$ marker)). In the figure 90% confidence ellipses are reported for each considered (scaling factor, encoding strategy) couple. Best results are obtained in the top right region of the figures (high SSIM and high bitrate reduction). Moreover,

¹Results obtained for the region at the right of the RoI are very close to those shown in Figure 2.4(b)

2. REDUCING THE NETWORK BANDWIDTH REQUIREMENTS FOR 360 IMMERSIVE VIDEO STREAMING

the smaller the confidence ellipses the more the strategy is insensitive to video diversity. Figure 2.4a shows that when the user's viewport is at the center of the RoI, SSIMs are always very high and close to the maximum. However, the higher the scaling factor, the higher the bitrate reduction, both in the case of the CRF and ABR cases. Let us now focus on Figure 2.4b showing the results when the user's viewport is at the left of the RoI, i.e. where the video has been downscaled.

Results show that the maximum scaling factor (SF=52%) for both ABR and CRF provides the best results in terms of bitrate reduction, but the consequent SSIM degradation is not negligible. Moreover, confidence ellipses are larger in the SSIM direction which indicates a higher sensitivity to video content. The best overall results are obtained for a scaling factor equal to 41% both for ABR and CRF: in particular, bitrate reductions are comparable to the ones obtained with scaling factor 52%, but the resulting SSIM is larger and confidence ellipses are considerably tighter. Therefore, the obtained results suggest that a scaling factor equal to 42% provides the best trade-off between visual quality and bitrate reduction.

2.6 Final considerations on the proposed codec-agnostic solution for bitrate reduction

In the previous Chapter, a codec-agnostic scheme to reduce network bandwidth requirements for immersive video streaming applications has been proposed. In summary, the major contribution is the idea is to downscale the areas outside the RoI, i.e. outside the current user's FoV, and to encode the resulting video. Then, the client decodes the video and upscales the video to the original resolution. The proposed scheme has been experimentally evaluated in order to measure the obtainable bitrate reductions. Moreover, the visual quality degradations due to downscaling have been estimated through the PSNR and SSIM metrics. Preliminary results show that it is possible to reach a bitrate reduction up to 50% while gracefully degrading the visual quality of regions of the video falling outside the RoI.

3

A DASH-compliant immersive streaming architecture

In this chapter a DASH-compliant video streaming control system for 360 immersive videos is described. The overall system is composed of two control algorithms which dynamically cooperate both to adapt the video bitrate to match the time-varying end-to-end network bandwidth and to select the most appropriate view of the panoramic scene in response to the varying point of view of the user. The proposed system has been implemented and an extensive experimental evaluation has been carried out in a realistic emulated network scenario to assess the obtainable performances in terms of visual quality and bitrate reduction. The scientific results discussed in the following chapter have been disseminated in [177] and more thoroughly in [178].

3.1 The Proposed Immersive Platform

Starting from the analysis of the State of the Art discussed in Chapter 1, it is possible to identify some different key logical operations for a general viewport-adaptive DASH-based Immersive Streaming platform, which are summarized in Figure 3.1.

In particular, the Server needs to provide the following logical operations:

- an agnostic encoding phase;
- a logical component which manages the creation of different viewpoint representations, where each representation embed a region of the represented omnidirectional scene where the perceptive quality is enhanced with respect to the other

3. A DASH-COMPLIANT IMMERSIVE STREAMING ARCHITECTURE

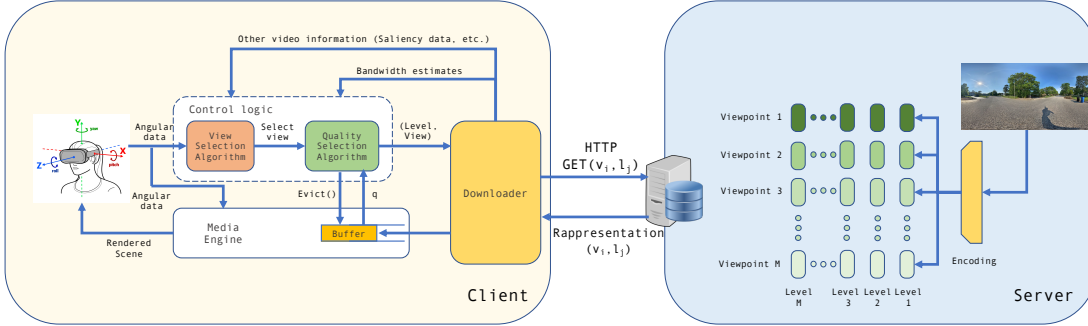


Figure 3.1: The viewport adaptive immersive streaming architecture

regions;

- a logical component which manages the creation of different quality representations from the set of viewpoint;

At Client, we need:

- a logical component performing the dynamic selection of the bitrate (i.e. of a MBR algorithm);
- a logical component performing the dynamic selection of the viewpoint representation based on the user needs;
- a media engine, who renders the 360 scene on the HMD.

3.1.1 DASH-compliant Server Design

In DASH systems the server is composed by a content generation algorithm and a storage system exposing the video segments to be downloaded.

To sum up, the idea is to generate from one projected video a number N of versions, the *views*, that encode RoIs at different α_i . All the views constitutes the *views set* $\mathcal{V} = \{v_1, \dots, v_N\}$. Now each view $v_i \in \mathcal{V}$ is encoded into M video representations at different bitrates (and resolutions) l_j constituting the video *level set* $\mathcal{L} = \{l_1, \dots, l_M\}$ with $l_j < l_{j+1}$. At the end of this procedure, the DASH Server stores and indexes a set of representations $\mathcal{R} = \mathcal{V} \times \mathcal{L}$ composed of $N \cdot M$ files.

As an example, in the Chapter 2 a real working content generation algorithm following the just mentioned guideline has been described.

3.1.2 Viewport adaptive Client

In DASH systems the client is composed by:

- the *media engine* - able to decode and render the downloaded 2D video chunks in a 3D virtual environment;
- the *control logic* - needed to dynamically select which video segment to download.

In particular, such control logic requires two cooperating components:

- a *quality selection algorithm* (QSA) that dynamically selects the video level $l(t) \in \mathcal{L}$ to avoid playback interruptions due to rebuffering while maximizing network channel utilization;
- a *view selection algorithm* (VSA) which dynamically chooses the most suitable view representation $v(t) \in \mathcal{V}$ to be downloaded based on measurements provided by the HMD accelerometer.

The QSA acts similarly to classic DASH adaptive video streaming algorithms (see for instance [48]). The VSA, aiming at selecting the best view representation depending on the current user's head position, is a new component that immersive video delivery systems have to implement. In the following paragraphs, further details on the QSA and the VSA logical controllers - respectively in Section 3.1.2.1 and in Section 3.1.2.2 - will be provided.

3.1.2.1 Quality Selection Algorithm

Concerning the QSA, its goal is to select the highest video level in such a way to adaptively match the time-varying network bandwidth, with the constraint of avoiding rebuffering events. In order to make clear how it works, the ELASTIC [48] adaptive control algorithm is described in the following.

The control law employed by ELASTIC to dynamically select the video level is defined as follows:

$$l(t_k) = \begin{cases} l(t_{k-1}) & q_L \leq q(t_k) \leq q_H \\ Q\left(\frac{b(t_k)}{1 - k_p e(t_k) - k_I e_I(t_k)}\right) & \text{otherwise} \end{cases} \quad \begin{matrix} (3.1) \\ (3.2) \end{matrix}$$

where $q(t_k)$ is the playout buffer length measured in seconds, $b(t_k)$ is the available bandwidth estimated at the end of the download of the k -th segment and $Q(\cdot) : x \mapsto l_i$

3. A DASH-COMPLIANT IMMERSIVE STREAMING ARCHITECTURE

is a quantizer function mapping the bitrate x to the closest video level bitrate $l_i \in \mathcal{L}$ which is lower than x . The error $e(t_k)$ is given by

$$e(t_k) = \begin{cases} q_L - q(t_k) & q(t_k) < q_L \\ q_H - q(t_k) & q(t_k) > q_H \\ 0 & \text{otherwise.} \end{cases}$$

Then, the cumulative sum $e_I(t_k)$ of the past values of the error $e(t_k)$ is defined as:

$$e_I(t_k) = \begin{cases} 0 & q_L \leq q(t_k) \leq q_H \\ \sum_k (t_k - t_{k-1})e(t_k) & \text{otherwise.} \end{cases}$$

In a nutshell, the algorithm works as follows: as long as the playout buffer length stays inside the playout buffer hysteresis ($q_L \leq q(t_k) \leq q_H$) the video level is kept constant (3.1) to contain the amount of video level switches which is known to have an adverse effect on the QoE. When the playout buffer gets outside the hysteresis, the controller sets the video level according to (3.2). Notice that (3.2) aims at steering the playout buffer length $q(t)$ towards the hysteresis when the playout buffer length gets outside of it. Thus, if the available bandwidth is roughly constant, it turns out that the queue is confined in the hysteresis and the video level will switch between the two adjacent levels which are closer to the available bandwidth. An important consequence of this property is that ELASTIC ensures that the average video level bitrate matches the average available bandwidth.

3.1.2.2 View Selection Algorithm

As stated before, the main goal of the VSA is to select the best viewpoint representation (a.k.a. *view* in the following) to be downloaded with the aim of maximizing the user's QoE.

The way the objective function for the QoE is undefined, but generally speaking it involves different decision variables such as for instance:

- current position of the user's head;
- segment duration;
- saliency data;

At this time, in order to provide a clear example of how the VSA works, a simple case where only the decision variable current user's head position is described in the

following. During the playback, the position of the head (i.e. the current yaw angle $\alpha(t)$) is continuously measured and checked against the set of yaw angles α_i associated to the different views (recall α_i is the angular position of the RoI center of i -th view).

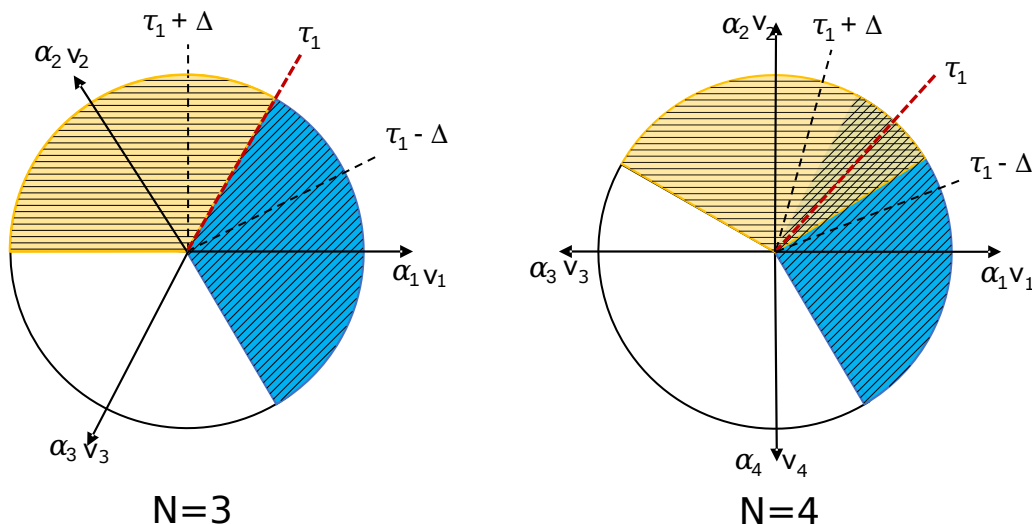


Figure 3.2: View selection algorithm in the case of $N = 3$ or $N = 4$ views

The VSA is designed to select the view that, based on the current position $\alpha(t)$, would provide to the user the largest high-quality area in the viewport. The approach is depicted in Figure 3.2 in the case of $N = 3$ or $N = 4$ views. The blue shaded area represents the RoI of the view v_1 , whereas the yellow shaded area corresponds to the view v_1 . For each view v_i a threshold τ_i is used to decide whether to switch to the view v_{i+1} . In particular, such thresholds are set as $\tau_i = (\alpha_{i+1} + \alpha_i)/2$. It is worth noting that in the case of $N = 4$ the views are partially overlapped. This case can result in a higher visual quality at end user, reducing the time spent in seeing regions at lower visual quality. In Section 3.2 further insights are provided.

Let us suppose the user is currently selecting view v_i . By employing such a setting for τ_i , it is very easy to check that when the user turns in a counterclockwise direction and $\alpha(t)$ surpasses τ_i , a switch to view $i + 1$ is needed to guarantee that the user enjoys the largest high-quality area in the viewport. In order to avoid unnecessary switches at the threshold boundary, an hysteresis centered on each τ_i is employed having an angular width equal to Δ degrees. The new optimal view is then chosen if the user crosses beyond such a threshold for more than K seconds, in order to limit the view

3. A DASH-COMPLIANT IMMERSIVE STREAMING ARCHITECTURE

switching frequency. When switching to a different view, to allow the user to perceive the improvement in visual quality within a reasonable time, a number of video frames stored in the buffer are evicted so that a configurable amount of playout buffer (named *safety margin*) is retained. This feature is extremely important because it allows to speed-up the visual improvement due to a view switch, while still allowing to avoid the occurrence of rebuffering events which are more likely to occur if the buffer occupancy is low.

3.2 Experimental Evaluation on a real Use Case

To assess the effectiveness of the proposed system in a real scenario, the immersive video delivery system described in Section 3.1 has been implemented.

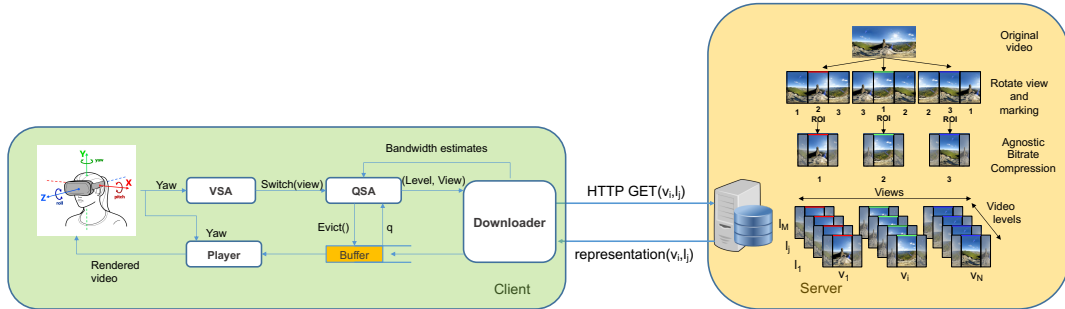


Figure 3.3: The proposed delivery system architecture

Figure 3.3 shows the overall architecture of the proposed delivery system which is composed of: a HTTP server the video content generation system (Section 3.1.1); a player running at the client which manages the control logic and the rendering of the received video (Section 3.1.2).

The content generation algorithm used to produce optimized immersive contents we choose to use the algorithm as described in 2.2.

In particular, on the one hand, a DASH Server (Debian Linux 9.12 workstation, 8GB RAM, Intel i7-4770 CPU @3.40GHz, running a 10.19.0 node version) has been set-up to implement the content generation algorithm described in Section 3.1.1, describing the video content in conformance to the specifications of the MPEG-DASH presentation format. On the other hand, the viewport-adaptive client designed in Section 3.1.2 has been developed as an HTML5 web player explicitly to be run on a common end-user

3.2 Experimental Evaluation on a real Use Case

laptop with Chromium web-browser. The web player makes use of standard technologies and open-source libraries to ensure compatibility with most modern browsers. The player exploits the WebGL-based open source library `THREE`¹ as the rendering engine. A specific mapping function (vertex shader) that properly associates the vertices of the mesh to the differently scaled strips of the video frame has been implemented to render the modified ERP produced by the content generation algorithm used at server side. The streaming engine has been built around the well-known open-source video streaming library `Shaka-player`². Both the QSA and the VSA have been implemented in Shaka as plugins. The player has been also modified to introduce additional features, including the support for the adopted custom MPD and the ability to allow partial evictions from the video buffer. The QSA controller keeps track of both estimated bandwidth and playout buffer length choosing the best *bitrate level* to download, while the VSA choose the best *view* in accordance to the head position of the user. The decision is taken during the download phase.

An extensive experimental campaign has been conducted to assess the performance gains offered by the proposed approach with respect to the usage of an adaptive streaming delivery system employing only one view identified in the following as the *baseline* approach. The classic dumb-bell network topology is employed. In particular, the client and the DASH server are connected through a bottleneck link with dynamically configurable capacity and latency through the `Mahimahi` tool [179]. To run the experiments, the 4K video sequence “Elephants on the Brink (360 Video)”³ was considered and the video player was instrumented to reproduce a realistic head movement according to the traces made available in [180]. The video has been encoded with different number of views ($N = 3$ or $N = 6$), downscaling resolutions (480px, 720px), and target video level bitrates as shown in Table 3.1. It is worth to note that in the case with $N = 6$ the resulting views contain RoIs partially overlapped.

The target bitrates for different parameter combinations have been chosen to ensure the same visual quality in the RoI region. This means that the visual quality of the RoI at a specific level l_i does not depend on the number of views and on the downscaling resolution of the regions falling outside the RoI. The 4K video sequence has been encoded

¹<https://threejs.org>

²<https://github.com/google/shaka-player>

³<https://youtu.be/2bpIC1A1g>

3. A DASH-COMPLIANT IMMERSIVE STREAMING ARCHITECTURE

N	Strategy	Downsc. res. [px]	l_1 (720p)	l_2 (1080p)	l_3 (2160p)
1	Baseline	1280	2.8 Mbps	5.2 Mbps	10 Mbps
3 or 6	N -480px	480	1.75 Mbps	3.25 Mbps	6.25 Mbps
	N -720px	720	2.1 Mbps	3.9 Mbps	7.5 Mbps

Table 3.1: Parameters used to encode the video levels and corresponding encoding bitrates

	ATT-LTE-driving-2016	ATT-LTE-driving	TMobile-LTE-driving	Verizon-LTE-short
Average	5.2249	7.4910	9.0846	4.6143
(DevStd)	(0.0931)	(0.1015)	(0.3065)	(0.3757)
[Mbps]				

Table 3.2: Average Bandwidth and Standard Deviation for each considered network trace

at 30 frames per second (fps) using the H.264 codec. The 4K video sequence has been segmented using the MP4 container, with two different segment durations, respectively 1.6s and 3.2s, and a group of picture (GOP) equal to the frame-rate multiplied by the segment duration, resulting in key-frames time-aligned across different levels and video segments. To evaluate the performance of the delivery systems considering different network conditions, four different mobile network traces (ATT-LTE-driving-2016, ATT-LTE-driving, TMobile-LTE-driving, Verizon-LTE-short) have been used, made publicly available by the MahiMahi suite¹. For each trace, the average bandwidth along with the corresponding standard deviation is shown in Table 3.2. Moreover, the performance of the system has been evaluated by considering two different safety margins, respectively equal to 5s or 3.2s. Notice that draining the buffer to a lower margin may increase the likelihood of incurring in playback interruptions, but could also lead to an improvement of the visual quality in the viewport due to the higher responsiveness of the system reacting to user’s head movements.

¹<http://mahimahi.mit.edu/>

3.3 Experimental Results

	3-480px	3-720px	6-480px	6-720px
1.6 s	21.9441 [2.7425] %	16.5618 [2.3358] %	24.1109 [5.9881] %	17.3631 [8.5262] %
3.2 s	11.7694 [4.6136] %	14.8192 [5.4837] %	18.9760 [6.3733] %	11.8218 [10.9257] %

Table 3.3: Average and standard deviation percentage of the reduction of segments bitrate in the case of segments duration respectively equals to 1.6s and 3.2s, for each considered network trace.

3.3 Experimental Results

In this section, the results obtained by employing the proposed approach integrated in a real-world DASH immersive adaptive streaming system are presented. Since the evaluation has not pointed out significant performance differences when employing a safety margin equal to 5s, in the following only the results obtained in the case of the queue safety margin set equal to 3.2s are reported.

Table 3.3 shows the average and standard deviation percentage of the reduction of the downloaded video segments bitrate with respect to the *baseline* case. In the case of segment duration equals to 1.6s, it clearly shows that the reduction of segments bitrate is around 22%(17%) when the downscaling resolution is set to 480px(720px): this is due to the lower target bitrate used to encode the video levels in the multiview case. Slightly lower performances are shown when a longer segment duration (set to 3.2s) is used: in this case, the reduction of segments bitrate is around to 14% in each considered case.

To better clarify this result, let us make a concrete example. The difference between the target bitrate for the 2160p level respectively in the *baseline* case and the $N = 3$ views case is 3.75 Mbps, that is the maximum bitrate reduction reachable in the case the available bandwidth is large and the user movement are reduced (a low number of view switches is produced). However, when the user’s head position changes and triggers a view-switch, the eviction of frames from the playout buffer can lead to downloading segments of the new view that were already downloaded for the previous view, thus lowering the reachable bitrate reduction. This issue is slightly more evident in the case of larger segment duration (3.2s) compared to the case of shorter segments (1.6s).

3. A DASH-COMPLIANT IMMERSIVE STREAMING ARCHITECTURE

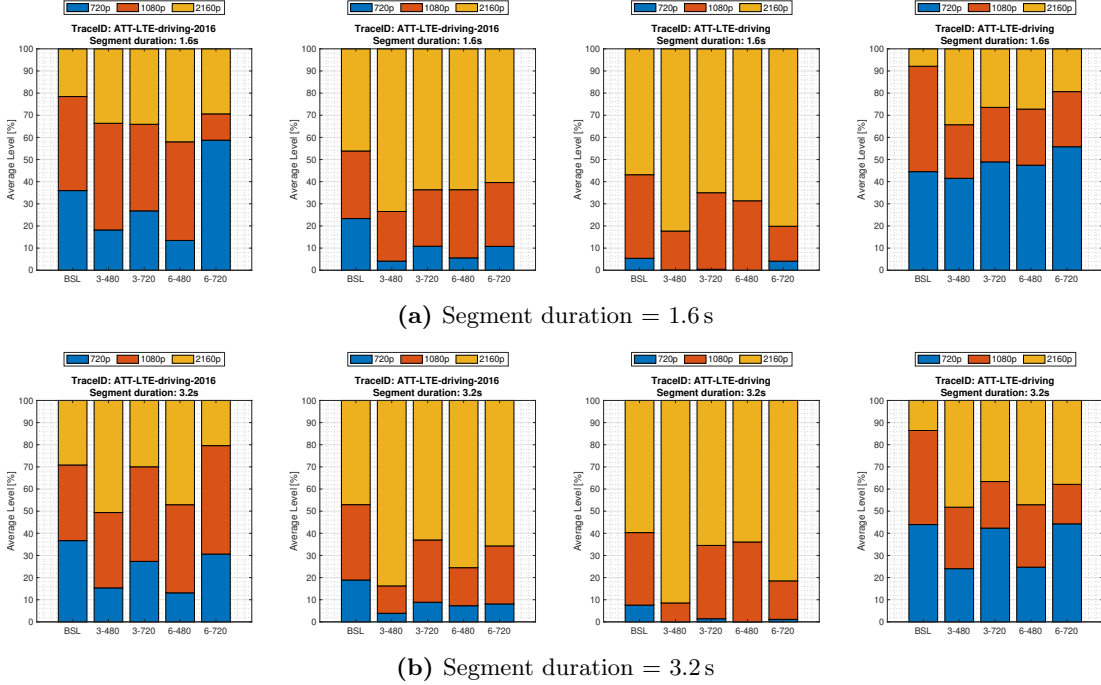


Figure 3.4: Breakdown of obtained video levels for each considered network trace

Figure 3.4 shows eight graphs grouped by each of the considered bandwidth traces, in the cases the segment duration is set respectively equal to 1.6 s (Figure 3.4a) or 3.2 s (Figure 3.4b). The graphs show the breakdown (expressed in percentage) of the visualized video levels for each considered video set parameter combination. In Figure 3.4, BSL represents the baseline case, while the case N-480 and N-720 (with N = 3 or 6) represent the streaming of N views, and the downscaled side is wide 480(or 720) pixels, respectively.

A larger percentage of higher resolution levels corresponds to a better experienced visual quality. Figure 3.4 shows that the percentage of segments with higher level (2160p and 1080p) in the case of multiple views is larger than that obtained in the baseline case. This is due to the relative smaller target bitrate used to encode the video levels in the case of 3 or 6 view with respect to the baseline case. This result confirms that the proposed approach allows to improve the visual quality compared to the baseline case in any of the considered bandwidth traces and parameter settings. Notice that the best results are obtained in the case of $N = 3$ with a downscaled resolution equal to 480p followed by the case of $N = 6$ with downscaled resolution

480p. Overall it is not surprising that performances obtained for downscaled resolution equal to 480p are better compared to the 720p case. This is due to the fact that video levels corresponding to the 480p case can be encoded at a lower target bitrate compared to the 720p downscaled resolution case. Notice that we have measured a negligible performance increase using $N = 6$ instead of $N = 3$. In general, having a large number of views N leads to an increased frequency of view switches which then triggers more frequent buffer evictions and then an increased rebuffering probability. This result, together with the increased storage costs required for the additional views, shows that it is advisable to keep the number of views low. Surprisingly, the performance advantage in using a shorter segment duration (Figure 3.4a) with respect to longer segment duration (Figure 3.4b) is negligible. This confirms the guess that the lower performance on bitrate reduction shown for the segment duration equals to 3.2s is due to the frame eviction mechanism.

Finally, in terms of rebuffering avoidance, the conceived streaming system has exhibited negligible rebuffering events (average around ~ 0.5 events per video streaming session) independently from the employed strategy parameters. It must be stressed that the obtained results are roughly the same for the baseline case, indicating that the proposed solution does not worsen the performance in terms of rebuffering. This result is due to the robustness of the ELASTIC quality selection algorithm, known to provide very low rebuffering ratios [48].

3.4 Final considerations

In the previous Chapter, a complete DASH-compliant Immersive Delivery Solution has been presented. The Immersive Streaming System was designed to be immediately deployable using existing hardware platforms and Internet infrastructures. The experimental performance evaluation was carried out emulating a mobile network and shown that the proposed DASH System improves the obtained average visual quality while providing rebuffering ratios close to zero in each of the considered scenarios, which together concurs to improve the overall QoE for the streaming of 360 videos.

3. A DASH-COMPLIANT IMMERSIVE STREAMING ARCHITECTURE

4

Bitrate Reduction for Immersive Streaming: Comparing Variable Quantization Parameter (VQP) and Variable Resolution (VRES) Approaches

In this Chapter, a brief description about the two approaches used in literature for implementing bitrate reduction of spatially partitioned immersive videos is provided. Moreover, an extensive performance evaluation has been carried out, and the most interesting results are presented.

4.1 Introduction

As seen in Section [1.5.2](#), the Tiling technique is a general technique which relies on the spatial portioning of the entire depicted scene, where each portion is manipulated in such a way to produce several representations with different resulting perceptive quality. The way to produce the each quality representation is left unspecified. Moreover, as discussed in Section [1.6](#), the Tile-based viewport-dependent streaming of 360 contents requires, on the one hand, the delivering at the final user of such tiles falling into its viewport with the highest possible quality; on the other hand, due to minimum final

4. BITRATE REDUCTION FOR IMMERSIVE STREAMING: COMPARING Variable Quantization Parameter (VQP) AND Variable Resolution (VRES) APPROACHES

bitrate requirements, the rest of the other tiles with the minimum possible quality. Moreover, sets of tiles can be arranged server-side for minimizing the number of GET required [140]. The result is the creation of different viewpoint representations.

It is worth noting that the algorithm used for producing the quality representation for each tile is unspecified. In general, different quality representations can be obtained for a generic video in two ways:

- by scaling the video itself at different resolutions;
- by compressing the video with different output bitrates.

In section 2 we discussed about a technique allowing to reduce the bitrate requirements for a viewpoint representation by reducing the resolution of the regions falling outside the desired RoI. The same objective can be obtained also by reducing the output bitrate for the same regions. In the following, a performance evaluation for the two aforementioned techniques is provided, highlighting their features and drawbacks.

4.2 Related Works

A performance evaluation of the 3D-to-2D projection methods is provided in [181]. In this work, several of the most commonly used projection functions are tested against different encoder implementations. Performances are measured in resulting quality, output bitrate and encoding efficiency (time). The results reveal that ERP grants the best resulting quality / bitrate ratio, while CMP shows better encoding efficiency. It is worth to remark here that the 3D-to-2D projection functions require being applied before encoding, requiring to modify the existing camera hardware and software to be efficient.

A different approach not requiring modifications to the existing encoders – named *divide-and-conquer* – is investigated in [167]. In summary, the idea is to divide the 360 scene in different spatial portions (*slice*). Each *slice* is then encoded independently and packaged separately in a different bitstream. Only the one falling in the current user’s FoV is delivered to the user. The advantage of this approach is the implementation simplicity, however, the drawback is that a RoI may span multiple slices, requiring one (hardware or software) decoding process *per slice* running on the client device. Moreover, the client has to download in parallel each slice composing the RoI, making the adaptive streaming algorithm considerably more complex. To solve this issue, in

[166] the authors take advantage of the HEVC tiling feature to implement a divide-a-conquer approach. The HEVC tiling feature allows to identify different spatial regions in a video and to set encoding parameters specific for that region. The resulting bitstream can be decoded with a single decoder instance at the client-side. Moreover, the authors in [182] [183] propose a HEVC tile-based 360° streaming framework as an Android application.

In [174], the authors use a multi-scale technique to add viewport-adaptivity to the 360° video and evaluate the proposed approach with respect to the offset projection and the tiling technique. The results show similar quality performances for multi-scale and tiling approaches, outperforming the offset strategy. However, the proposed multi-scale encoding strategy is quite complex and can be hardly used for realtime streaming. In Section 2.2 (with the scientific results disseminated in [131]) an encoder-agnostic technique leveraging the RoI concept for reducing the bitrate requirements of the 360° video has been described. In particular, the goal is reached by properly downsampling the spatial regions outside the identified RoI.

4.3 Bitrate Reduction Techniques

Figure 4.1 shows the pipeline used to produce the OV content which is divided into four parts.

In the *RoI selection* phase an algorithm detects a higher interest area spanning 120 horizontally. The algorithm used to select the most interesting areas can be a general *content-aware* algorithm based on saliency map, such as the one described in [184]. This way multiple views can be produced each on centered at a specific RoI. The *projection* phase projects the entire 3D sphere onto a 2D plane using the ERP-format. Notice that each area of the 360 video, i.e. the RoI, the region at the Left (L) and at its Right (R), corresponds to a vertical strip of equal horizontal resolution res_0 in the ERP projection.

The *encoding* and *decoding* phases differ depending on the approach used to reduce the bitrate. In the following, we separately describe the two considered bitrate reduction approaches and summarize their main advantages and drawbacks.

4. BITRATE REDUCTION FOR IMMERSIVE STREAMING: COMPARING Variable Quantization Parameter (VQP) AND Variable Resolution (VRES) APPROACHES

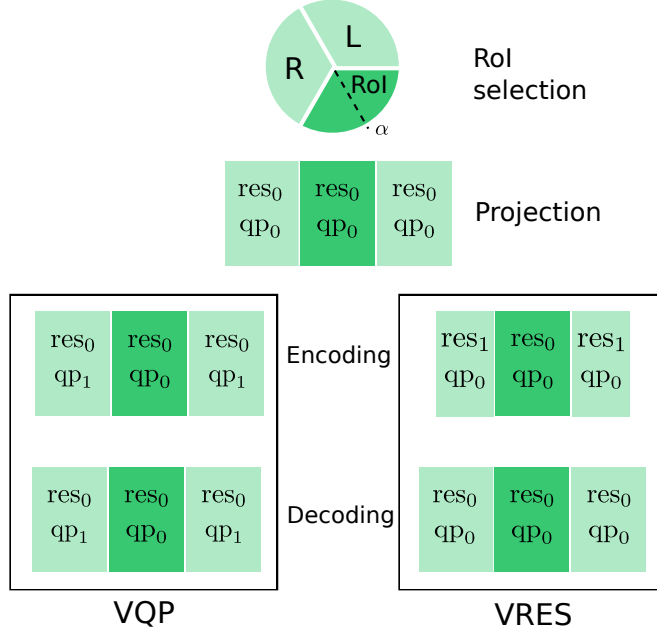


Figure 4.1: A sketch of the streaming pipeline under test.

4.3.1 VQP approach

Let us start by describing the *encoding* phase in the case of the VQP approach, shown in the left branch of Figure 4.1. In the encoding phase the resolution of the three regions is kept unchanged to res_0 . Each region is then mapped in a different MCTS, thus allowing the decoding process being fully parallelizable. The encoder quantization parameter is set to qp_0 in the RoI region, whereas the regions outside the RoI (L and R) are encoded at a higher quantization parameter equal to $qp_1 = qp_0 + \Delta qp$. In the decoding phase, no particular operation is needed to be performed in the case of the VQP approach: decoding is then performed in parallel by a single HEVC decoding instance for all the three downloaded tiles.

This approach allows performing deep server-side storage optimization techniques (such as the user-centric server optimisation proposed in [184]). Nevertheless, as stated in Section 2.1, a drawback of this approach is that is strictly bounded to the HEVC, thus requiring specific hardware support for decoding not widely available in the mobile market at the moment [131].

4.3.2 VRES approach

The VRES approach is shown in the right branch of Figure 4.1. In this case, the encoding phase requires that the two regions outside the RoI are shrunk horizontally from a resolution res_0 to a lower resolution res_1 . Next, the resulting rescaled video is encoded at a quantization parameter equal to qp_0 applied to all the ERP video. After the video is decoded, the two regions outside the RoI are upscaled from res_1 to the original horizontal resolution res_0 .

With respect to the VQP approach, this technique has interesting features: 1) it is independent of the employed codec, 2) it can be efficiently handled by hardware decoders at the client-side, 3) it can use well-established and mature algorithms (interpolation, filtering, etc.) to improve the resulting video quality. Nevertheless, server storage consumption can be high if RoI selection phase is not appropriately tuned.

4.4 Methodology

Table 4.1 lists the video catalog used for the performance comparison. All the considered videos have a resolution of 3840×1920 and a framerate of 30 fps. To consider the set of settings commonly used for online streaming, the GOP parameter was fixed to 150, which means that a key-frame is generated every 5 seconds. The visual quality assessment between the manipulated video and the reference one has been obtained by means of the visual quality metric VMAF [185] which has proven to be effective also for 360 videos [186]. Notice that the dataset also includes SSIM scores which however prove less expressive compared to the ones obtained using VMAF and therefore are not discussed in the following.

The visual quality assessment for each video has been carried out as described in the following. For each video in the catalog (assumed as the reference video), a manipulated copy has been produced according to the considered bitrate reduction strategy, namely VQP and VRES. Both the manipulated and the reference video have been segmented at the GOP boundaries, producing a chunk set with chunks duration equal to 5 seconds. An area, centered at *yaw_angle* and wide horizontally 120, has been cropped for each chunk from both the manipulated and the reference video chunk set. The two cropped areas have been evaluated with the VMAF visual quality metric to produce a score. The *yaw_angle* varies in the set $\{-120, -100, -90, \dots, 90, 100, 120\}$ to cover the entire

4. BITRATE REDUCTION FOR IMMERSIVE STREAMING: COMPARING Variable Quantization Parameter (VQP) AND Variable Resolution (VRES) APPROACHES

Table 4.1: The video catalog used in the test

Video	Youtube ID
Boomerang	r-qmDDi8S5I
FighterJet	NdZ02-Qenso
UniversalStudiosFlorida	Js_Jv5Ez0v0
Tahiti360	7gjR60TSn8Q
KITZ360	KS9S1Hgx2co
WhiteLions360	1407AxqjiVY
WildDolphins	BbT_e81WWdo
GirlGroup360	NxIRVu110CA
MaldivesVR360	MgJITGvVfR0

360 field of view. Notice that the *yaw_angle* equal to 0 corresponds to the case in which only the RoI (that is never degraded) is framed in the viewport. The extreme case where the user frames in the viewport only degraded content corresponds to either *yaw_angle*=-120 or 120.

The VRES and VQP approaches have been tested leveraging the tiling feature implemented by the *kvazaar* encoder [172]. The *kvazaar* encoder allows to set the grid to be used to divide the video in tiles. To comply with the rationale used in [177], a 3-column grid as been applied each having horizontal resolution equal to 1280p. The `--mv-constraint frametilemargin` option usage ensures that the encoder operations to be fully parallelizable for each tile, by managing the HEVC MTCS feature. Furthermore, in the case of the VQP approach, the encoder enables to specify the variation of the quantization parameter (Δqp) to be applied to each tile with respect to a baseline quantization parameter qp_0 . In the experiments, the Δqp varies in the set {5, 10, 15, 20}. Notice that the lower Δqp the lower is the expected bitrate reduction.

The VRES approach implements the bitrate reduction strategy as described in [177]. Again, to provide a fair performance evaluation the same encoder, i.e., *kvazaar* is used to encode the same video catalog. In this case, the `--mv-constraint frametilemargin` option has been left unset. The VRES approach has been tested with four different

downscaled resolutions res_1 , namely 1080p, 720p, 480p, 240p.

As already mentioned above, videos encoded with the VRES approach need to upscale the encoded video to the original resolution. Such an operation is performed through an interpolator filter. To the purpose, in this work we have employed the bicubic interpolator made available by the FFmpeg suite.

To investigate the relationship between the obtainable bitrate reduction and the resulting video quality, the encoder have been set in Constant Quality (CQ) mode. When configured in this mode, the encoder is free to vary the output bitrate to reach the set output video quality. In this work, the `--qp` parameter has been chosen to output a visually lossless video quality. Moreover, it is worth to remark here that we are interested on bitrate reduction capability of the algorithms, not on absolute output bitrate. As reported in [1], the `--qp` value has been set equal to 22, i.e., for VRES the whole video is compressed with $\text{qp}_0 = 22$. In the case of VQP the RoI is encoded at $\text{qp}_0 = 22$, whereas the regions falling outside of the RoI are encoded with a quantization parameter equal to $\text{qp}_0 + \Delta\text{qp}$.

Finally, the obtained dataset comprises around 64,000 VMAF scores obtained by analyzing a total of around 88 hours of video content. Also notice that the entire duration of the videos has been analyzed.

4.5 Results

This section presents the obtained results and it is organized as follows. We first show the impact of the parameters used by the two approaches on the obtained bitrate reduction (Section 4.5.1). Then, we compare the overall visual quality obtained by each of the considered approaches as a function of the position of the users' head (Section 4.5.2). We next delve into investigating how the video content impacts the differences between the visual quality obtained by the VRES and VQP approaches 4.5.3.

4.5.1 Bitrate reduction

We start our investigation by considering the efficiency in terms of bitrate reduction of VRES and VQP schemes as a function of their respective parameters. In particular,

¹<https://github.com/ultravideo/kvazaar>

4. BITRATE REDUCTION FOR IMMERSIVE STREAMING: COMPARING Variable Quantization Parameter (VQP) AND Variable Resolution (VRES) APPROACHES

for VRES the rescaled resolution varies in $\{1080p, 720p, 480p, 240p\}$, whereas in the case of VQP Δqp varies in $\{5, 10, 15, 20\}$.

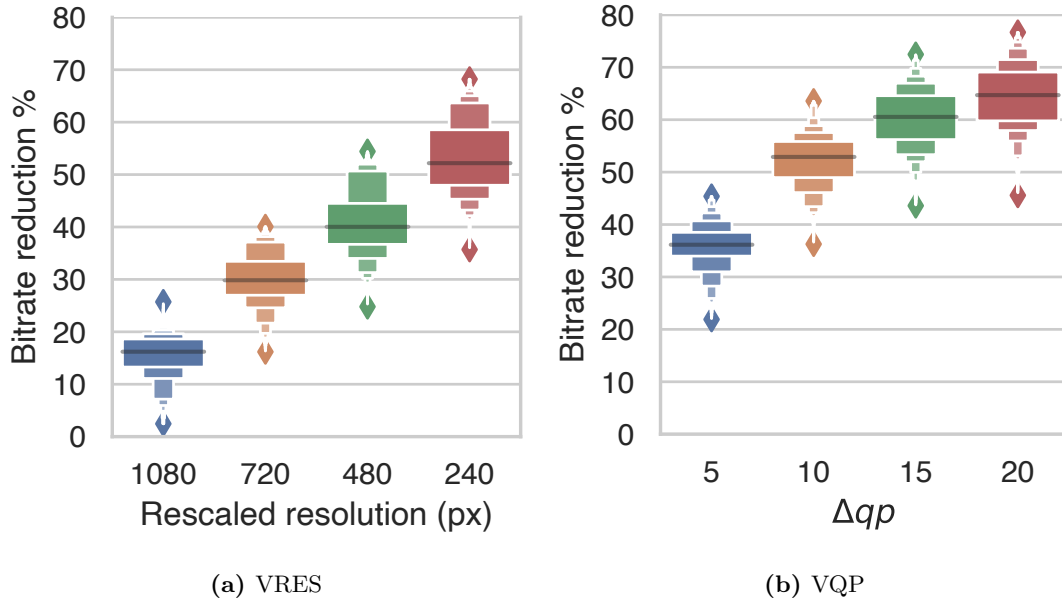


Figure 4.2: Bitrate reduction (%)

Figure 4.2a and Figure 4.2b compare the overall percentage of bitrate reduction which can be obtained by both VRES and VQP considering the whole video catalog. The results are shown in a boxplot which captures the variability of the results with respect to different videos and segment in the video.

The Figure 4.2a shows that, in the VRES case, as the rescaled resolution decreases from 1080p to 240p, the obtained bitrate reduction increases from a median value of around 15% up to 52% quite linearly. Regarding the VQP approach, Figure 4.2b shows that the impact of Δqp on bitrate reduction is more pronounced as this parameter increases. In particular, $\Delta qp = 5$ already provides a median bitrate reduction of around 36%, and rapidly increases to 52% for $\Delta qp = 10$ which is exactly equal to the maximum bitrate reduction obtained in the case of the VRES approach when the rescaled resolution is set to 240p. Also, comparable median bitrate reductions are obtained for $\Delta qp = 5$ (corresponding to $\sim 36\%$) and for rescaled resolution 480p (corresponding to 40%).

In the next sections, we shall employ the established couples of parameters that

provide similar bitrate reductions, i.e. (5, 480p) and (10, 240p), to compare the corresponding visual quality obtained.

4.5.2 Visual quality as a function of the user’s head position

We are now interested in comparing the visual quality obtained by VRES and VQP when they offer comparable bitrate reductions. To the purpose, for each video we collect the VMAF score measured when the users’ head is positioned at a certain yaw angle α with respect to the center of the RoI. Recall that, $\alpha = 0$ corresponds to the case in which the viewport only frames the 120-wide area that is non distorted. As α moves away from the RoI, larger and larger degraded portions of the video will fall in the users’ viewport and the visual quality is expected to decrease. In the case of VQP, the degradation is due to the higher quantization parameter used to encode the content outside the RoI, in the VRES case, the degradation is due to the downscaling and upscaling operations described in Section III and IV.

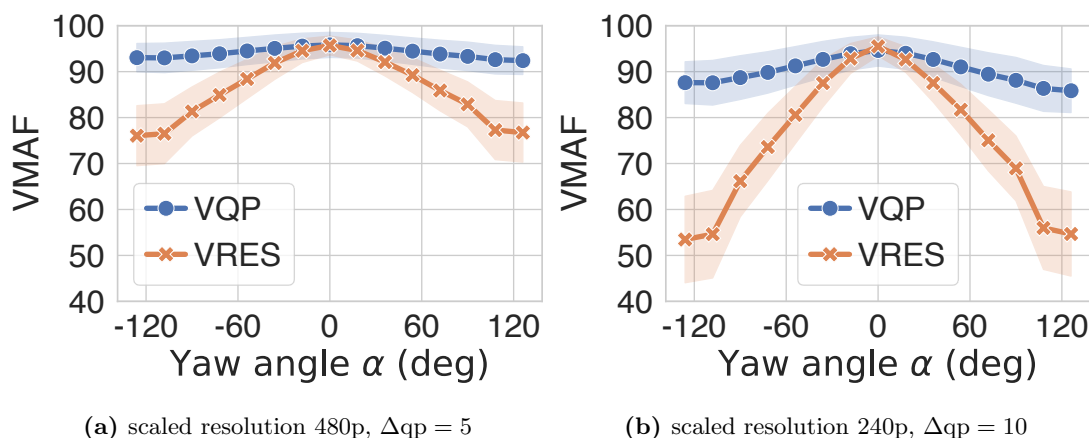


Figure 4.3: VMAF as a function of the user’s head yaw angle α

Figure 4.3a and Figure 4.3b compare the median visual quality and the standard deviation (shaded areas) measured using the VMAF score as a function of the yaw angle α . Let us start by considering Figure 4.3a which corresponds to the case in which VQP employs a $\Delta qp = 5$ to encode the regions outside the RoI and VRES downscales the horizontal resolution of the regions outside the RoI to 480p. In Section 4.5.1, we have shown that those parameters provide a comparable median bitrate reduction of around 40%. Figure 4.3a shows that, as expected, as $|\alpha|$ increases the measured

4. BITRATE REDUCTION FOR IMMERSIVE STREAMING: COMPARING Variable Quantization Parameter (VQP) AND Variable Resolution (VRES) APPROACHES

VMAF decreases. Nevertheless, in the case of the VQP approach the quality degrades negligibly, whereas in the VRES case the VMAF drops from ~ 95 ($\alpha = 0$) to ~ 76 ($\alpha = 120^\circ$)¹.

Figure 4.3b compares the case of VQP set with a $\Delta qp = 10$ and VRES set with a downscale resolution equal to 240p which corresponds to a median bitrate reduction of around 52% for both the approaches. The figure confirms that VQP is able to provide a graceful degradation of the visual quality obtaining a worst case VMAF score equal to ~ 85 , whereas VRES achieves a worst case measured VMAF as low as ~ 52 . This means that in the VRES case if users point their head to a region framing only distorted content the obtained visual quality is between “poor” and “fair” [185].

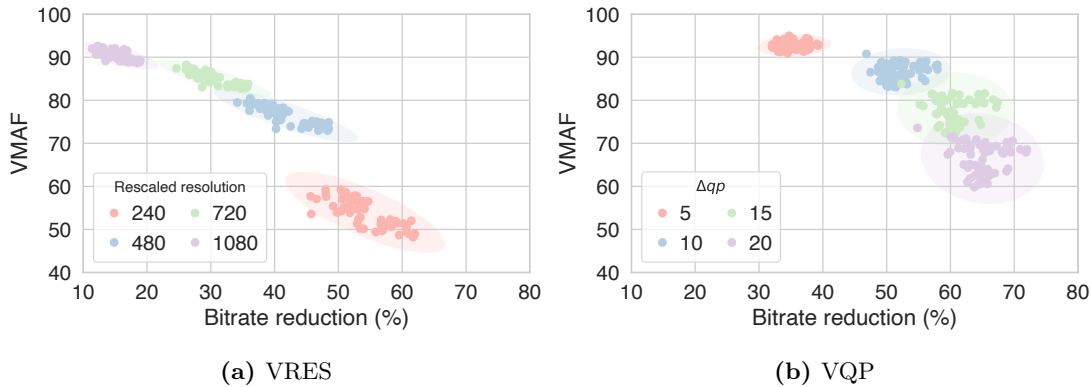


Figure 4.4: Worst case Visual Quality vs Bitrate reduction trade-off

To complete this analysis, Figure 4.4a and Figure 4.4b show the worst case VMAF bitrate reduction trade-off achieved respectively by VRES and VQP obtained when $\alpha = 120^\circ$. Each data point of the scatter plot represents one video chunk of a given video encoded with a specific parameter (differentiated by its color). The interesting insight that can be gathered from the Figure 4.4b is that, in the case of VQP, increasing the Δqp parameter from 15 to 20 increases the bitrate reduction negligibly (as pointed out in Section 4.5.1) at the price of a drastic decrease of the worst case visual quality from a median value of ~ 80 to ~ 65 .

In summary, VRES visual quality decreases faster when the user moves his head away from the RoI, whereas VQP gracefully degrades the visual quality. For VQP using

¹According to VMAF authors a score equal to 70 can be mapped to a vote between “good” and “fair” [185].

4.6 Final considerations about VRES and VQP bitrate reduction schemes

a Δ_{qp} greater than 15 is not advisable.

4.5.3 Visual quality as a function of video content

In Section 4.5.2 we have found that, in the worst case, the median difference between the VMAF score of VQP and VRES is equal to ~ 16 (~ 30) when the bandwidth reduction percentage is $\sim 40\%$ (52%) (see Figure 4.3a and Figure 4.3b). In this section, we are interested in investigating the sensitivity of the two bitrate reduction strategies to different video content. To the purpose, Figure 4.5a and Figure 4.5b compare the VMAF scores for each content of the video catalog (see Table 4.1) in the worst case when the yaw angle is equal to 120° .

The figures show that in 7 out of 9 videos the VMAF scores do not differ significantly from the median value we have found in Section 4.5.2. Nevertheless, the video *WhiteLions360* shows a remarkably lower VMAF score in the case the VRES strategy is used, whereas in the case of *WhiteDolphins* the VMAF scores are much closer with respect to the median case.

Figure 4.6 shows one frame extracted from the *WhiteLions360* at a yaw angle such that the left half of the frame belongs to the RoI, whereas the right half of the frame belongs to the distorted area outside of the RoI. The parameters employed for VRES and VQP lead to a 52% bandwidth reduction. By comparing the two frames it can be noticed that: i) in the VRES case the gaussian blur effect is clearly visible on the lion; this is due to the lossy process of downscaling the region outside the RoI from 1280p to 240p and then re-upscaling the video to the original resolution; ii) in the VQP case the frame is sharp also in the region where the higher quantization parameter is used (compare the field texture and the leaves of the tree); nevertheless, some artifacts affect the frame in the degraded region which are clearly visible on the lion's face and mane.

4.6 Final considerations about VRES and VQP bitrate reduction schemes

In this chapter, we compared the two State-of-the-Art (SOTA) bitrate reduction schemes: the VRES approach and the VQP approach using the kvazaar encoder. To the purpose, we have measured the VMAF score and the obtained bitrate reduction percentage by applying both VRES and VQP approaches to a catalog of nine benchmark 4K resolution

4. BITRATE REDUCTION FOR IMMERSIVE STREAMING: COMPARING Variable Quantization Parameter (VQP) AND Variable Resolution (VRES) APPROACHES

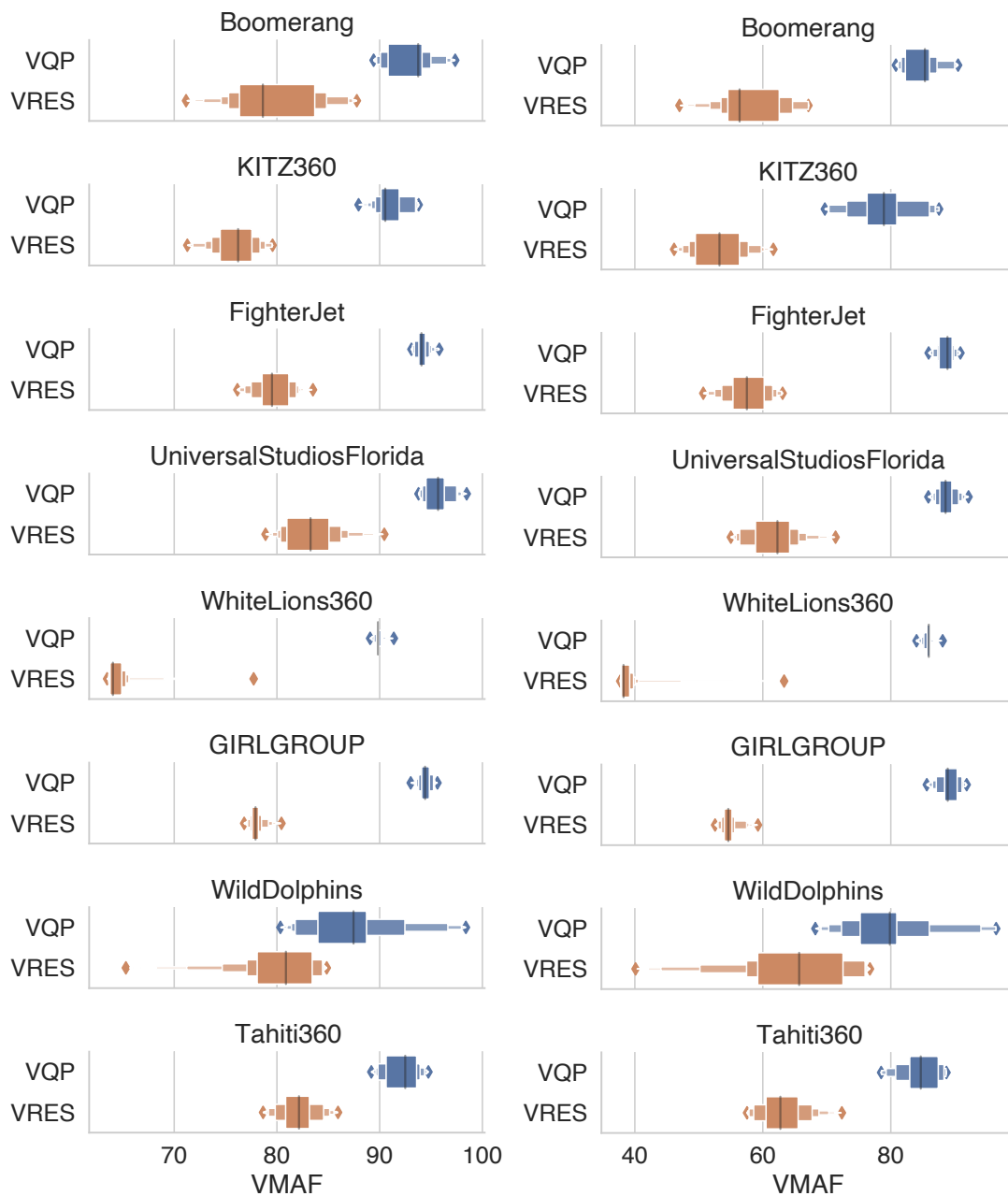


Figure 4.5: Worst case Visual Quality vs Bitrate reduction trade-off for each video

4.6 Final considerations about VRES and VQP bitrate reduction schemes



(a) VQP, $\Delta qp = 10$

(b) VRES, rescaled resolution 240p

Figure 4.6: Frame extracted from the WhiteLions360 video

videos. Results have shown that at equal bitrate reduction percentage the VMAF score obtained by VRES is consistently lower than that of VQP. When the two approaches achieve a bitrate reduction percentage equal to 52%, the VQP obtains a VMAF scores higher up to 30 points compared to VRES. Nevertheless, at lower bitrate reductions (i.e., when the rescaled resolution is near to 480p), the VRES approach does not pay a remarkable quality loss and becomes a viable solution due to its implementation simplicity and due to the fact that it can be employed with any codec.

**4. BITRATE REDUCTION FOR IMMERSIVE STREAMING:
COMPARING Variable Quantization Parameter (VQP) AND Variable
Resolution (VRES) APPROACHES**

5

TAPAS-360: a Tool for the Design and Experimental Evaluation of 360 Video Streaming Systems

In this Chapter I present TAPAS-360, an open-source tool that enables designing and experimenting all the components required to build omnidirectional video streaming systems. The tool can be used by researchers focusing on the design of viewport-adaptive algorithms and also to produce video streams to be employed for subjective and objective QoE evaluations. The TAPAS-360 presented in this chapter has been described in the scientific research work [\[187\]](#).

5.1 Introduction

Video streaming platforms are required to innovate their delivery pipeline to allow new and more immersive video content to be supported. In particular, Omnidirectional Video (OV) enable the user to explore a 360 scene by moving their heads using HMD devices. Viewport adaptive streaming allows changing dynamically the quality of the video falling in the user's FoV. Experimental research in this area requires building a full pipeline which starts from immersive content generation and ends at video consumption using a player. All these components must implement the required features to make the

5. TAPAS-360: A TOOL FOR THE DESIGN AND EXPERIMENTAL EVALUATION OF 360 VIDEO STREAMING SYSTEMS

interaction with the 360 scene possible. If the research community has proposed several tools for the design and experimental evaluation of Adaptive BitRate (ABR) algorithms [188, 189], the same cannot be said about omnidirectional videos. As a result, it is quite difficult to reproduce the results of different viewport adaptive algorithms and to make fair comparison among such algorithms.

In our previous work [188], we proposed a TAPAS a framework allowing the researcher to only concentrate on the design of the ABR algorithm without the need of implementing a complete player for classic 2D adaptive streaming. Building on the core functionalities of TAPAS, this work presents TAPAS-360, a tool which significantly extends TAPAS and allows rapid prototyping of viewport adaptive control algorithms used for the distribution of immersive content. The tool has been designed to decrease the computational load required for each video stream generated on the testing machine. In particular, since panoramic video decoding is the process having the greatest impact on the computational load, TAPAS-360 allows to optionally disable the video decoding process while keeping the dynamics of the playout buffer, and therefore of the overall video streaming session, unchanged. Consequently, it is possible to carry out accurate experiments involving a large number of concurrent flows using the same machine. This feature is fundamental for experimentally studying the performance of the video distribution system as the number of streams that insist on the same link changes. Moreover, the tool can be easily used in combination with common network emulation tools such as, f.i., MahiMahi¹ to perform experiments in a controlled network environment, allowing the reproducibility of the obtained results. Additionally, traces of head movement [180, 190] can be used to experimentally evaluate the performances of the viewport adaptive algorithms with respect to different viewing patterns, or to test field-of-view prediction algorithms. At the best of our knowledge, there are no open-source tools available that implement the features described above. TAPAS-360 currently supports only viewport-adaptive schemes that download the whole sphere, whereas supporting schemes that download only portions of the whole sphere is a planned feature to be implemented in the future.

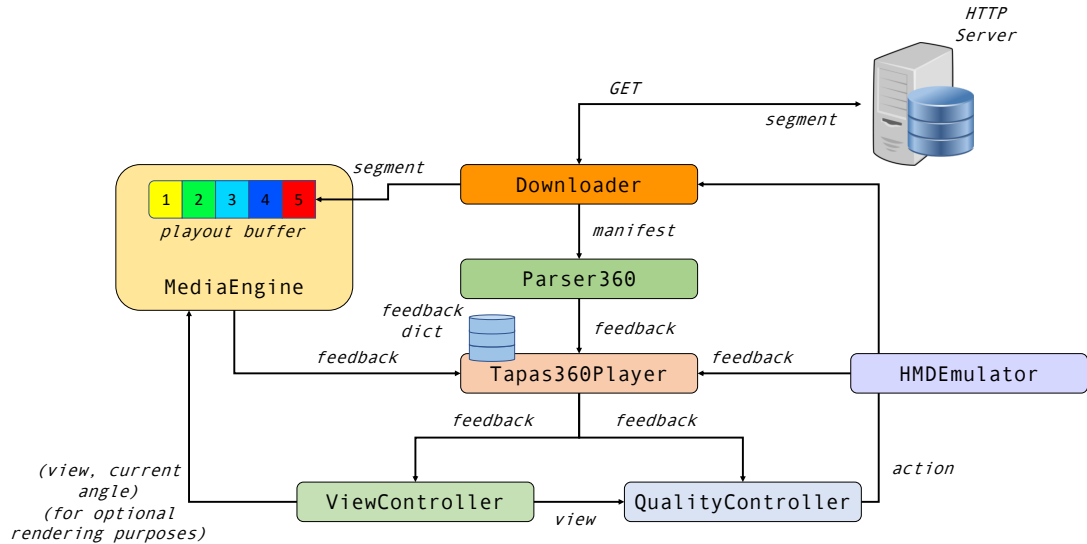


Figure 5.1: Workflow of the TAPAS-360 tool

5.2 TAPAS-360

Figure 5.1 shows a block diagram of the TAPAS-360 tools highlighting the main components and the corresponding connections between them. In addition to the features of rapid prototyping, flexibility of use, and modularity inherited from the predecessor [188], TAPAS-360 integrates a whole series of new modules allowing the management of immersive content compliant with the DASH SRD specifications [132]. In the following, the essential details of the components is provided, whereas specific implementation details are left to the documentation of the project available in the project repository.

5.2.1 Tapas360Player

Tapas360Player is the central module that deals with orchestrating the operations of all TAPAS-360 components. This module implements the player logic and updates the log files that are populated during the experiments and that can be used in the post-processing phase for analyzing the performance of the implemented algorithms. The communication between modules is implemented through the exchange of a **feedback**

¹<http://mahimahi.mit.edu/>

5. TAPAS-360: A TOOL FOR THE DESIGN AND EXPERIMENTAL EVALUATION OF 360 VIDEO STREAMING SYSTEMS

dictionary. This dictionary contains all the pieces of information that are useful for performing the experiments and to pass data from one module to the other.

The `play` method is used to start the experiment and manages the user interaction, initializes the feedback dictionary, and orchestrates the operational flow of the various modules composing TAPAS-360.

5.2.2 Parser360

The main task of this module is to retrieve and store information about the video manifest. It performs the parsing operation required for the particular streaming standard (HLS or DASH) in use. `Parser360` populates and keeps updated the `playlists` data structure. In this data structure, each segment is identified by the relative URI. Moreover, the piece of information about the particular *level* (or representation), together with the respective parameters such as resolution and bitrate, are stored to allow the required control actions being performed. In the case of immersive content, additional information about the possibly different viewpoint representations is also stored. At the end of the update process, the `playlists` data structure is passed to the `Tapas360Player` module for updating the `feedback` dictionary: in the case of *live streaming*, the `playlists` data structure is continuously updated, while in the case of *video on demand* (VoD) the data structure is populated once at the startup.

The implementation of a parser requires the extension of the `BaseParser360` class and the definition of two methods: `start()`, which retrieves and analyzes the manifest to populate the `playlists` data structure and `updateSegmentsList()`, which keeps updated the `playlists` structure.

5.2.3 MediaEngine

`MediaEngine` is the module dealing with the management of the playback operation. In details, it is responsible for maintaining the playout buffer, providing optional features for the possible decoding and rendering of the video stream. `BaseMediaEngine` provides the skeleton class defining the following methods: `start()`, `stop()`, `pushData()` and `getQueuedTime()`.

Going into detail, the `start()` method initializes the playout buffer and the other components required by the specific `MediaEngine` implementation. `MediaEngine` allows the configuration of the parameter `min_queue_time` that is the minimum duration of

video stored in the buffer to start the video playback. Each time a video segment download is completed, the `pushData()` callback is called and the segment is pushed to the playout buffer.

The `getQueuedTime()` method allows to read the length of the playout buffer measured in seconds. This information is useful for the ABR controller. The `Tapas360Player` module uses this method to allow updating the feedback dictionary.

The specific implementation of `MediaEngine` must extend `BaseMediaEngine`. This way, different logics for draining the buffer, decoding and playing the video stream can be defined. To allow different degrees of simulation details, two multimedia engines have been implemented in TAPAS-360: `GstMediaEngine` and `FakeMediaEngine`. `FakeMediaEngine` emulates the player status by tracking the length of the playout buffer based on the information contained in the incoming segments without demuxing nor decoding the received video stream. Instead, `GstMediaEngine` provides a complete multimedia engine. Based on *GStreamer 1.0*^[1] multimedia framework, it is able to manage both *fMP4* and *ts* media formats, granting compatibility with HLS and DASH streaming standards. Moreover, it can work into two modes: `nodec` mode, only demuxing the incoming stream flow; `dec` mode, with video stream decoding and rendering capabilities.

Both `FakeMediaEngine` and `GstMediaEngine` modules allow to disable the video decoding process to keep CPU and memory usage low while perfectly emulating the dynamics of the playout buffer, therefore having the same overall system dynamics as in the case where the received video stream is decoded and rendered. This is a key feature that enables to experimentally study the performance of the video distribution system as the number of streams that share the same bottleneck varies. Moreover, `GstMediaEngine` in `dec` mode can decode, render and possibly store the rendered video stream on the filesystem (see Section 4.3).

5.2.4 QualityController

The `QualityController` is the module responsible for implementing the ABR algorithm. Its goal is to decide, based on feedback information such as the estimated bandwidth, the length of the playout buffer, and the status of the player, which video representation to download from those listed in the *manifest*.

¹<https://gstreamer.freedesktop.org/>

5. TAPAS-360: A TOOL FOR THE DESIGN AND EXPERIMENTAL EVALUATION OF 360 VIDEO STREAMING SYSTEMS

`BaseQualityController` provides the interface class that a `QualityController` must inherit. Important methods that have to be implemented are: 1) `calcControlAction()`, that implements the control logic by calculating the maximum bitrate value that should be downloaded; 2) `isBuffering()`, that checks if the player is currently into downloading (*buffering* phase) or in *idle* phase (used to insert OFF pauses between the download of two consecutive segments).

To clarify how the `QualityController` module works, the salient logical sequence of operations that `Tapas360Player` implements is reported. `Tapas360Player` maintains a `feedback` dictionary which stores various information such as the length of the playout buffer and the estimated bandwidth. At the end of the download of each segment, `Tapas360Player`, using the `updateSegmentsList()` method exposed by `Parser360` class, updates the `feedback` dictionary and executes `calcControlAction()` to obtain the video level to be used for the download of the next segment and sets the period of inactivity by using `setIdleDuration()`. In particular, `calcControlAction()` returns the maximum bitrate value that can be downloaded based on the information contained in the `feedback` dictionary. This value is then passed to `quantizeRate()` that selects the highest video level index from the possible values contained in the `feedback` dictionary. In its default implementation, the `quantizeRate()` method selects the highest video level lower the bitrate calculated by `calcControlAction()`. Finally, the `isBuffering()` method checks if the system is either buffering or idle. This is a useful method to keep track of the player state and manage rebuffering events. `BaseQualityController` provides a default implementation for this method, returning `True` if the length of the playout buffer is less than a certain threshold, but more advanced mechanisms can be implemented by overloading this method.

5.2.5 ViewController

The `ViewController` is a new component that immersive video streaming systems are required to implement. Its goal is to select the best viewpoint representation according to the position of the user's head which is reported by the HMD device.

The implementation of the `ViewController` needs to extend the `BaseViewController` class and to implement the `getView()` method, which actually defines the viewpoint selection algorithm.

An example implementation of view controller, named `ConventionalViewController` is included in the code base which provides a simple control logic that takes as input the current position of the user’s head. The `ConventionalViewController` implements the View Selection Algorithm (VSA) described in [177]. In that paper, different viewpoint representations are prepared server-side, each one consisting in a different *Region of Interest* (RoI), the particular region of the video where the visual quality is higher with respect to the other regions. Each viewpoint representation is identified by a URL and correlated to a tuple storing the *identifier* and the yaw angle pointing to the corresponding RoI. The VSA goal is to select the best viewpoint representation based on the current user view direction.

`ConventionalViewController`, in its current implementation, assumes RoIs are centered at 0, 120 and 240 with a dihedral angle of 120, resulting in three different *tiles set*. Nevertheless, *Saliency maps* could be used to tailor the selection of the number and position of the RoIs [190]. Notice that we plan to add support for saliency maps to be integrated in the base view controller class soon so that view controllers will be able to readily access this optional information.

The VSA algorithm workflow is described briefly in the following. The `getView()` method returns to `Tapas360Player` the viewpoint representation that the user is currently viewing. At the end of the download of each segment, `Tapas360Player` – through the `getHMDStatus()` method exposed by the `HMDEmulator` class – updates the `feedback` dictionary which also stores the current viewpoint and the angles representing the position of the user’s head. Next, it executes the `getView()` method which returns the viewpoint representation to be selected. At this point, the downloader automatically downloads the correct viewpoint representation and the current bitrate representation selected by the `QualityController`.

5.2.6 HMDEmulator

In TAPAS-360, the design of `ViewController` requires to receive in input the current angular position from an HMD to perform the viewpoint selection strategy. `HMDEmulator` is the module that emulates the reading of the angles of the user’s head position which normally are provided by the HMD device.

`HMDEmulator` implements the `getCurrentViewAngle()` method, which accepts the playback timestamp and returns the angular data of the current position of the user’s

5. TAPAS-360: A TOOL FOR THE DESIGN AND EXPERIMENTAL EVALUATION OF 360 VIDEO STREAMING SYSTEMS

head. Such information can be also exploited for viewport adaptive control algorithms based on saliency maps. The emulation of the user’s head movement is obtained by reading a Comma-Separated Values (CSV) file which contains the angular data relating to the user’s head movement at each timestamp of playback. In this way, publicly available datasets such as [180, 184, 190] can be easily used to allow result reproducibility.

5.3 Use Cases

In this section we describe some of the use cases for which TAPAS-360 has been designed for.

5.3.1 2D video streaming

The most common use case is the design and the development of new ABR strategies¹. To this end, only the `BaseController.py` class has to be extended, implementing the control logic in the `calcControlAction()` method. The new ABR algorithm can be added to `play.py` by simply importing it. The command line for testing the algorithm is:

```
$python3 play.py --controller [CONTROLLER] --url [URL]
```

where `[CONTROLLER]` is the name of the class containing the algorithm being tested and `[URL]` is the URL indicating the manifest of the testing video. TAPAS-360 will fetch the video segments indicated in `[VIDEO-URL]` as a regular video player would do under the bitrate adaptation algorithm implemented. The `logs/` folder contains a list of subfolders, indexed for streaming session, with all the logs useful for postprocessing and performance evaluation.

5.3.2 Viewport-adaptive streaming

The most interesting use case is the design and experimental evaluation of *viewport adaptive* algorithms. To this end, the developer can extend only the `BaseViewController.py` class. The `getView()` method implements the viewport adaptive strategy. Similarly to the `QualityController` algorithm, `play.py` has to be modified importing the

¹Notice that this use case was already possible with TAPAS [188], but we mention it for the sake of completeness.

new class and adding a custom entry into the flags list (if needed). To test the newly implemented algorithm the following command can be used:

```
$python3 play.py --vr True --view_controller [VIEWCONTROLLER]
--url [URL]
```

where [VIEWCONTROLLER] is the name of the class implementing the viewport adaptive algorithm and [URL] is the URL indicating the manifest of the testing video. Similarly to the ABR case, TAPAS-360 will perform the viewport adaptation strategy in the same way as a 360 video player would do. This is possible because of the `HMDemulator` is fed with a trace representing head movement¹ having the format [time, alpha, beta, gamma], where time is a timestamp and alpha, beta, gamma are the three components of the Euler angles. Custom HMD traces can be used in TAPAS-360 by employing the `--hmd_trace` flag. Also in this case useful logs are available into the `logs/` folder.

5.3.3 Subjective and Objective Quality of Experience evaluations

TAPAS-360 allows to store the fetched segments by simply adding the option

```
$python3 play.py [other_options] --save_chunks True
```

The list of the segments is stored in the subfolder corresponding to the streaming session under the `logs/` folder. This allows to produce video streams that can be used, together with the video streaming log, to run subjective and objective QoE evaluations. To this end, TAPAS-360 could be used to produce a number of “distorted” videos in response to both time-varying network bandwidths (implemented with tools such as `Mahimahi`) and head movements, by feeding TAPAS-360 traces from datasets such as [180, 184, 190].

5.4 Summary

In this Chapter the TAPAS-360 open-source tool has been presented. In summary, TAPAS-360 enables designing and experimenting streaming algorithms for Immersive Applications. The tool allows a fine grained control of the decoding process to significantly decrease the CPU load and enable experimenting with several flows being consumed on a single machine. TAPAS-360 includes extensible modules to experiment

¹Notice that a default example trace in the repository named `hmd_trace.csv` is provided.

5. TAPAS-360: A TOOL FOR THE DESIGN AND EXPERIMENTAL EVALUATION OF 360 VIDEO STREAMING SYSTEMS

with viewport adaptive algorithms and to emulate HMD devices using head movements datasets. The ambition is to attract the research community to contribute with their algorithms and make TAPAS-360 an open platform facilitating results reproducibility within the multimedia community.

6

Conclusions and Future Research Directions

This PhD thesis has investigated the Immersive streaming ecosystem, concerning the important aspects such as resource optimization against user QoE expectations.

First of all, the Thesis provides a thorough State-of-the-Art, ranging from the upcoming technologies used for streaming the traditional 2D video to the most recent advancements specific for the streaming of Immersive contents.

In particular, the today internet bandwidth has been recognized as insufficient for the streaming of immersive contents at a satisfactory QoE, thus the design of new bitrate reduction techniques exploiting the peculiarities of such contents is required. With reference to the wide set of techniques described in Section 1.5, in Chapter 2 a viewport-dependent technique aiming at reducing the network bandwidth requirements for immersive video streaming applications has been presented. In summary, the technique i) realizes bitrate reductions by aggressively reducing the horizontal resolution of the areas outside the main RoI; ii) is an encoder-agnostic approach; iii) adopts standard upscaling-downscaling algorithms, thus hardware encoding-decoding capabilities can be fully exploited. The performance of the proposed approach has been experimentally evaluated on a video catalog using both the well established PSNR and SSIM objective visual quality metrics. The experimental results show that the proposed approach is able to provide a reduction of the required bitrate up to around 50%, while gracefully degrading visual quality far from the user's RoI.

Moving the target on a system perspective, Chapter 3 provided a characterization of

6. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

a DASH-based Immersive Video Streaming platform, identifying the logical components which specifically manage the streaming of Immersive contents.

In summary, we identified two main components:

- a Server, equipped with a content generation algorithm specifically designed for immersive content optimization;
- a Client, which selects the best video representation to download based on the decisions of two cooperating control logics:
 1. the Quality Selection Algorithm (QSA), realizing the bitrate adaptation;
 2. the View Selection Algorithm (VSA), realizing the viewport adaptation.

To assess the effectiveness of the proposed system in a real scenario, the immersive video delivery system has been implemented and subjected to an extensive experimental evaluation. On the one hand, the technique introduced in Chapter 2 has been used at Server as a content generation algorithm. On the other hand, the Client implemented ELASTIC as QSA, while a naive control law based on current user head position has been utilized as VSA. In summary, we tested the proposed system with a number of views equals both three and six, with three downscaling factors for the non-RoI areas. More, we tested the platform against four network traces, using the MahiMahi shaper for simulating the varying network conditions. The baseline refers to the platform without viewport-adaptation. Briefly, we obtained promising results in the case of viewport-adaptivity is enabled.

In the context of bitrate reduction techniques, we proceeded with the State-of-the-Art and identified the two most promising techniques, named as:

- the Variable Quantization Parameter (VQP), which allows to vary the spatial quality without changing output resolution;
- the Variable Resolution (VRES), which allows to vary the spatial resolution without changing quality.

It is worth noting that the VRES approach is quite similar to the technique proposed in Chapter 2.

To assess the performances reachable by the two techniques in terms of maximal QoE, we go through a really extensive QoE comparison, which involved a dataset of more of 88 hours of video content. In this performance evaluation, we decided to use

the VMAF objective visual quality metric, that is now considered the state of the art for QoE video evaluation.

Finally, in Chapter 5 TAPAS-360, a tool aiming at being useful in designing and experimentally evaluating 360-degree streaming systems, has been proposed to the immersive streaming research community.

The motivation behind the development of this tool is quite straightforward: at this time, experimental research on 360 streaming systems requires to build a full pipeline which starts from content generation and ends at video consumption. Then, it is quite difficult for the researcher to reproduce the results of different research work and to make fair comparisons among the developed algorithms. The TAPAS-360 tool allows both the rapid prototyping and ease the task of experimentally comparing different viewport adaptive algorithms. The tool has been designed to decrease the computational load required for each video stream by disabling the decoding process without interfering with the dynamics of the streaming session, as an example the playout buffer dynamics. Consequently, it is possible to carry out massive experimentation involving a large number of concurrent flows on the same machine.

Finally, the tool can be easily extended with common network emulation tools (such as, for instance MahiMahi) to perform massive experimentations in a controlled network environment. In this way the reproducibility of the obtained results is implemented out of the box.

From the works previously discussed, it emerges that times are nearly mature for adopting Immersive streaming applications in everyday life. Specifically, viewport adaptive approaches have been recognized as an attractive solution, allowing the bitrate reduction for Immersive contents with a perceived QoE comparable to the case of no adaptation; or, on other words, allowing to deliver Immersive contents with an higher QoE on the todays internet connections.

As regarding future directions of these research activities, a deeper analysis on parallel encoding/decoding algorithms will be performed, with the aim of leveraging the dense multi-core architecture offered by upcoming GPUs. Moreover, 6-DoF volumetric videos present even higher bandwidth requirements with respect to 3-DoF omnidirectional video, thus new analysis will be made specifically for this kind of video format. A final future work will be to identify further use cases of Immersive technologies at the service of the Industry 4.0.

6. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

References

- [1] 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*, vol. 13, no. 4, pp. 600–612, 2004. [vii](#), [22](#), [23](#)
- [2] Z. Chen, Y. Li, and Y. Zhang, “Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation,” *Signal Processing*, vol. 146, pp. 66–78, 2018. [vii](#), [30](#), [31](#), [32](#), [35](#), [37](#)
- [3] Facebook developers, “Enhancing high-resolution 360 streaming with view prediction,” <https://code.fb.com/virtual-reality/enhancing-high-resolution-360-streaming-with-view-prediction/>, Apr 2017, online; accessed 02-Jul-2020. [vii](#), [37](#), [38](#)
- [4] Facebook developers, “Next-generation video encoding techniques for 360 video and VR,” <https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/>, Jan 2016, online; accessed 02-Jul-2020. [vii](#), [38](#), [39](#)
- [5] C. Zhou, Z. Li, and Y. Liu, “A measurement study of oculus 360 degree video streaming,” in *Proc. of the 8th ACM on Multimedia Systems Conference*, ser. Proc. of ACM MMSys ’17. New York, NY, USA: ACM, 2017, pp. 27–37. [viii](#), [39](#), [40](#)
- [6] C. V. N. Index, “Global mobile data traffic forecast update, 2016–2021,” *white paper*, 2017. [xi](#)

REFERENCES

- [7] M. Hosseini and V. Swaminathan, “Adaptive 360 VR video streaming based on MPEG-DASH SRD,” in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 407–408. [xi](#), [28](#), [42](#)
- [8] Netflix, “Internet Connection Speed Recommendations,” <https://help.netflix.com/en/node/306>, online; accessed 02-Jul-2020. [xii](#)
- [9] J. Thompson, J. Sun, R. Möller, M. Sintorn, G. Huston, and D. Belson, “Q1 2017 State of the Internet-Connectivity Report,” *Akamai, Tech. Rep.*, 2017. [xii](#)
- [10] Y. S. de la Fuente, G. S. Bhullar, R. Skupin, C. Hellge, and T. Schierl, “Delay impact on MPEG OMAF tile-based viewport-dependent 360 video streaming,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 18–28, 2019. [xii](#)
- [11] M. L. Heilig, “Sensorama simulator,” Aug. 28 1962, uS Patent 3,050,870. [1](#)
- [12] I. E. Sutherland, “The ultimate display,” *Multimedia: From Wagner to virtual reality*, vol. 1, 1965. [2](#)
- [13] I. E. Sutherland, “A head-mounted three dimensional display,” in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 1968, pp. 757–764. [2](#)
- [14] L. Furness, “The application of head-mounted displays to airborne reconnaissance and weapon delivery,” *Wright-Patterson Air Force Base, Ohio, USA*, 1969. [3](#)
- [15] D. F. Kocian, “A visually-coupled airborne systems simulator (vcass)-an approach to visual simulation,” AIR FORCE AEROSPACE MEDICAL RESEARCH LAB WRIGHT-PATTERSON AFB OH, Tech. Rep., 1977. [3](#)
- [16] J. J. Batter and F. P. Brooks Jr, “GROPE-1: A computer display to the sense of feel,” in *IFIP Congress (1)*, 1971, pp. 759–763. [3](#)
- [17] M. W. Krueger, T. Gionfriddo, and K. Hinrichsen, “VIDEOPLACE - an artificial reality,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1985, pp. 35–40. [3](#)

-
- [18] B. L. Burnside, “Assessing the capabilities of training simulations: A method and Simulation Networking (SIMNET) application,” ARMY RESEARCH INST FOR THE BEHAVIORAL AND SOCIAL SCIENCES ALEXANDRIA VA, Tech. Rep., 1990. [3](#)
- [19] M. W. McGreevy, “The virtual environment display system,” 1991. [3](#)
- [20] P. Milgram and F. Kishino, “A taxonomy of mixed reality visual displays,” *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994. [3](#), [4](#)
- [21] T. Mazuryk and M. Gervautz, “Virtual reality-history, applications, technology and future,” 1996. [3](#)
- [22] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, “The CAVE: audio visual experience automatic virtual environment,” *Communications of the ACM*, vol. 35, no. 6, pp. 64–73, 1992. [3](#)
- [23] M. Heim, *Virtual realism*. Oxford University Press, 2000. [3](#)
- [24] R. Carey, “Virtual Reality Modeling Language (VRML97),” *ISO/IEC 14772-1: 1997*, 1997. [4](#)
- [25] W. Consortium *et al.*, “Extensible 3D (X3D), ISO/IEC 19775-1: 2008,” *Visited on*, 2008. [4](#)
- [26] M. Billinghurst, A. Clark, and G. Lee, “A survey of augmented reality,” 2015. [4](#)
- [27] J. Vince, *Virtual reality systems*. Pearson Education India, 1995. [5](#)
- [28] J. Schild, D. Lerner, S. Misztal, and T. Luiz, “EPICSAVE - Enhancing vocational training for paramedics with multi-user virtual reality,” in *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, 2018, pp. 1–8. [6](#)
- [29] B. Byl, M. Süncksén, and M. Teistler, “A serious virtual reality game to train spatial cognition for medical ultrasound imaging,” in *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*. IEEE, 2018, pp. 1–4. [6](#)

REFERENCES

- [30] K. Chen, Z. Chen, Y. Tai, J. Peng, J. Shi, and C. Xia, "A System Design for Virtual Reality Visualization of Medical Image," in *2018 26th International Conference on Geoinformatics*. IEEE, 2018, pp. 1–5. [6](#)
- [31] M. K. Bekele, R. Pierdicca, E. Frontoni, E. S. Malinverni, and J. Gain, "A survey of augmented, virtual, and mixed reality for cultural heritage," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 11, no. 2, pp. 1–36, 2018. [6](#)
- [32] U. Luna, P. Rivero, and N. Vicent, "Augmented reality in heritage apps: Current trends in Europe," *Applied Sciences*, vol. 9, no. 13, p. 2756, 2019. [6](#)
- [33] V. Vlahakis, J. Karigiannis, M. Tsotros, M. Gounaris, L. Almeida, D. Stricker, T. Gleue, I. T. Christou, R. Carlucci, N. Ioannidis *et al.*, "Archeoguide: first results of an augmented reality, mobile computing system in cultural heritage sites," *Virtual Reality, Archeology, and Cultural Heritage*, vol. 9, no. 10.1145, pp. 584 993–585 015, 2001. [6](#)
- [34] E. Bonacini, "La realtà aumentata e le app culturali in Italia: storie da un matrimonio in mobilità/Augmented reality and cultural apps in Italy: stories on a marriage in mobility," *Il capitale culturale. Studies on the Value of Cultural Heritage*, no. 9, pp. 89–121, 2014. [6](#)
- [35] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for Internet video streaming," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2530–2543, 2000. [6](#)
- [36] D. McNamee, C. Krasic, K. Li, A. Goel, E. Walthinsen, D. Steere, and J. Walpole, "Control challenges in multi-level adaptive video streaming," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, vol. 3. IEEE, 2000, pp. 2228–2233. [6](#)
- [37] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Internet Requests for Comments, RFC Editor, STD 64, July 2003, <http://www.rfc-editor.org/rfc/rfc3550.txt>. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3550.txt> [7](#)

-
- [38] T. Friedman, R. Caceres, and A. Clark, “RTP Control Protocol Extended Reports (RTCP XR),” Internet Requests for Comments, RFC Editor, RFC 3611, November 2003. [7](#)
- [39] H. Schulzrinne, A. Rao, and R. Lanphier, “Real Time Streaming Protocol (RTSP),” Internet Requests for Comments, RFC Editor, RFC 2326, April 1998, <http://www.rfc-editor.org/rfc/rfc2326.txt>. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc2326.txt> [7](#)
- [40] A. Yaqoob, T. Bi, and G. Muntean, “A Survey on Adaptive 360 Video Streaming: Solutions, Challenges and Opportunities,” *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2020. [7](#), [8](#), [29](#), [54](#)
- [41] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” Internet Requests for Comments, RFC Editor, STD 86, July 2017. [7](#)
- [42] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009, pp. 1–12. [7](#)
- [43] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hofffeld, and P. Tran-Gia, “A survey on quality of experience of HTTP adaptive streaming,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2014. [8](#)
- [44] G. Carlucci, L. De Cicco, and S. Mascolo, “HTTP over UDP: an Experimental Investigation of QUIC,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 609–614. [8](#)
- [45] G. Cofano, L. De Cicco, and S. Mascolo, “Modeling and design of adaptive video streaming control systems,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 548–559, 2016. [8](#), [51](#)
- [46] R. van Brandenburg, O. van Deventer, F. Faucheur, and K. Leung, “Models for HTTP-Adaptive-Streaming-Aware Content Distribution Network Interconnection (CDNI),” *Internet Engineering Task Force, Informational RFC*, vol. 6983, 2013. [8](#)

REFERENCES

- [47] L. De Cicco, S. Mascolo, and V. Palmisano, “Feedback control for adaptive live video streaming,” in *Proc. of 2nd ACM Conference on Multimedia Systems*, ser. MMSys ’11, 2011, pp. 145–156. [9](#), [59](#)
- [48] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, “ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH),” in *2013 20th International Packet Video Workshop*, Dec 2013, pp. 1–8. [9](#), [50](#), [52](#), [69](#), [77](#)
- [49] R. Pantos and W. May, “HTTP Live Streaming,” Internet Requests for Comments, RFC Editor, RFC 8216, August 2017. [9](#)
- [50] ISO/IEC, “ISO/IEC 23009-1: 2014: Information technology-Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats,” 2014. [9](#), [10](#), [12](#)
- [51] S. Zhao, Z. Li, and D. Medhi, “Low delay MPEG DASH streaming over the WebRTC data channel,” in *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2016, pp. 1–6. [10](#)
- [52] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over HTTP,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 325–338. [10](#), [53](#)
- [53] D. O. Trujillo, G. E. C. Golondrino, D. F. D. Dorado, W. Y. C. Muñoz, and J. L. A. Herrera, “Coding multimedia content using DASH standard,” in *2016 IEEE 11th Colombian Computing Conference (CCC)*. IEEE, 2016, pp. 1–7. [10](#)
- [54] I. DASH, “Guidelines for Implementation: DASH-IF Interoperability Points, Version 3.1,” in *DASH Interoperability Forum*. [12](#)
- [55] D. Singer, “ISO/IEC 14496-12: 2005 part 12: Iso base media file format,” *International Organization for Standardization*, 2005. [13](#)
- [56] I. Rec, “H. 222.0 - ISO/IEC 13818-1,” *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Systems*, ITU-T/ISO/IEC, 2007. [13](#)

-
- [57] B. Rainer, S. Lederer, C. Müller, and C. Timmerer, “A seamless Web integration of adaptive HTTP streaming,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 1519–1523. [13](#)
- [58] ISO/IEC, “ISO/IEC 223000-19:2020 2014: Information technology-Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media,” 2020. [13](#)
- [59] Apple, “About the Common Media Application Format with HTTP Live Streaming,” https://developer.apple.com/documentation/http_live_streaming/about_the_common_media_application_format_with_http_live_streaming, online; accessed 02-Jul-2020. [14](#)
- [60] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, “Vr is on the edge: How to deliver 360 videos in mobile networks,” in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 30–35. [15](#), [29](#), [40](#)
- [61] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012. [15](#), [17](#)
- [62] T. Wiegand, “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H. 264— ISO/IEC 14496-10 AVC),” *JVT-G050*, 2003. [15](#)
- [63] I. Rec, “H. 265 and ISO/IEC 23008-2: High efficiency video coding,” *ITU-T and ISO/IEC JTC*, vol. 1, p. 20, 2013. [15](#)
- [64] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, “Block partitioning structure in the HEVC standard,” *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1697–1706, 2012. [15](#), [16](#)
- [65] D. Patel, T. Lad, and D. Shah, “Review on intra-prediction in high efficiency video coding (HEVC) standard,” *International Journal of Computer Applications*, vol. 975, no. 8887, p. 12, 2015. [16](#)

REFERENCES

- [66] R. Sjöberg, Y. Chen, A. Fujibayashi, M. M. Hannuksela, J. Samuelsson, T. K. Tan, Y.-K. Wang, and S. Wenger, “Overview of HEVC high-level syntax and reference picture management,” *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1858–1870, 2012. [16](#), [17](#), [45](#)
- [67] K. Misra, A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou, “An overview of tiles in HEVC,” *IEEE journal of selected topics in signal processing*, vol. 7, no. 6, pp. 969–977, 2013. [18](#)
- [68] A. K. Moorthy, K. Seshadrinathan, R. Soundararajan, and A. C. Bovik, “Wireless video quality assessment: A study of subjective scores and objective algorithms,” *IEEE transactions on Circuits and Systems for Video Technology*, vol. 20, no. 4, pp. 587–599, 2010. [19](#), [22](#)
- [69] P. ITU-T RECOMMENDATION, “Subjective video quality assessment methods for multimedia applications,” *International telecommunication union*, 1999. [19](#), [20](#)
- [70] T. Hofffeld, M. Hirth, J. Redi, F. Mazza, P. Korshunov, B. Naderi, M. Seufert, B. Gardlo, S. Egger, and C. Keimel, “Best practices and recommendations for crowdsourced qoe-lessons learned from the qualinet task force” crowdsourcing”,” 2014. [21](#)
- [71] M. Cheon and J.-S. Lee, “Subjective and objective quality assessment of compressed 4k uhd videos for immersive experience,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 7, pp. 1467–1480, 2017. [21](#)
- [72] A. M. Rohaly, P. J. Corriveau, J. M. Libert, A. A. Webster, V. Baroncini, J. Beerends, J.-L. Blin, L. Contin, T. Hamada, D. Harrison *et al.*, “Video quality experts group: Current results and future directions,” in *Visual Communications and Image Processing 2000*, vol. 4067. International Society for Optics and Photonics, 2000, pp. 742–753. [21](#)
- [73] S. A. Mahmood and R. F. Ghani, “Objective quality assessment of 3d stereoscopic video based on motion vectors and depth map features,” in *2015 7th Computer Science and Electronic Engineering Conference (CEEC)*. IEEE, 2015, pp. 179–183. [21](#)

-
- [74] T.-J. Liu, Y.-C. Lin, W. Lin, and C.-C. J. Kuo, “Visual quality assessment: recent developments, coding applications and future trends,” *APSIPA Transactions on Signal and Information Processing*, vol. 2, 2013. [21](#), [22](#)
- [75] M. T. Vega, V. Sguazzo, D. C. Mocanu, and A. Liotta, “Accuracy of no-reference quality metrics in network-impaired video streams,” in *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia*, 2015, pp. 326–333. [21](#)
- [76] B. Girod, “What’s wrong with mean-squared error?” *Digital images and human vision*, pp. 207–220, 1993. [22](#)
- [77] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, “Image quality assessment based on a degradation model,” *IEEE transactions on image processing*, vol. 9, no. 4, pp. 636–650, 2000. [22](#), [24](#)
- [78] A. Nasrabadi, M. Shirsavar, A. Ebrahimi, and M. Ghanbari, “Investigating the psnr calculation methods for video sequences with source and channel distortions,” in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. IEEE, 2014, pp. 1–4. [22](#)
- [79] Q. Huynh-Thu and M. Ghanbari, “Scope of validity of psnr in image/video quality assessment,” *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008. [22](#)
- [80] A. B. Watson, Q. J. Hu, and J. F. McGowan, “Digital video quality metric based on human vision,” *Journal of Electronic imaging*, vol. 10, no. 1, pp. 20–30, 2001. [22](#)
- [81] H. Fletcher, “Physical measurements of audition and their bearing on the theory of hearing,” *The Bell System Technical Journal*, vol. 2, no. 4, pp. 145–180, 1923. [22](#)
- [82] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402. [23](#)

REFERENCES

- [83] Z. Wang and E. P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2. IEEE, 2005, pp. ii–573. [23](#)
- [84] C. Li and A. C. Bovik, "Three-component weighted structural similarity index," in *Image quality and system performance VI*, vol. 7242. International Society for Optics and Photonics, 2009, p. 72420Q. [23](#)
- [85] Z. Wang and Q. Li, "Information content weighting for perceptual image quality assessment," *IEEE Transactions on image processing*, vol. 20, no. 5, pp. 1185–1198, 2010. [23](#)
- [86] A. Rehman, K. Zeng, and Z. Wang, "Display device-adapted video quality-of-experience assessment," in *Human Vision and Electronic Imaging XX*, vol. 9394. International Society for Optics and Photonics, 2015, p. 939406. [23](#)
- [87] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *Journal of electronic imaging*, vol. 19, no. 1, p. 011006, 2010. [24](#)
- [88] D. M. Rouse, R. P epion, S. S. Hemami, and P. Le Callet, "Image utility assessment and a relationship with image quality assessment," in *Human Vision and Electronic Imaging XIV*, vol. 7240. International Society for Optics and Photonics, 2009, p. 724010. [24](#)
- [89] S. Li, F. Zhang, L. Ma, and K. N. Ngan, "Image quality assessment by separately evaluating detail losses and additive impairments," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 935–949, 2011. [24](#)
- [90] T.-J. Liu, W. Lin, and C.-C. J. Kuo, "Image quality assessment using multi-method fusion," *IEEE Transactions on image processing*, vol. 22, no. 5, pp. 1793–1807, 2012. [24](#)
- [91] J. Y. Lin, T.-J. Liu, E. C.-H. Wu, and C.-C. J. Kuo, "A fusion-based video quality assessment (fvqa) index," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 2014, pp. 1–5. [24](#)

-
- [92] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock, “Vmaf: The journey continues,” *Netflix Technology Blog*, 2018. [25](#)
- [93] I. Katsavounidis, “Dynamic optimizer—a perceptual video encoding optimization framework,” *The Netflix Tech Blog*, 2018. [25](#)
- [94] M. Manohara, A. Moorthy, J. De Cock, I. Katsavounidis, and A. Aaron, “Optimized shot-based encodes: Now streaming,” *The Netflix Tech Blog*. [Online] Available: <https://medium.com/netflix-techblog/optimized-shot-based-encodes-now-streaming-4b9464204830>, vol. 2, 2018. [25](#)
- [95] R. Rassool, “Vmaf reproducibility: Validating a perceptual practical video quality metric,” in *2017 IEEE international symposium on broadband multimedia systems and broadcasting (BMSB)*. IEEE, 2017, pp. 1–2. [25](#)
- [96] C. Lee, S. Woo, S. Baek, J. Han, J. Chae, and J. Rim, “Comparison of objective quality models for adaptive bit-streaming services,” in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. IEEE, 2017, pp. 1–4. [25](#)
- [97] N. Barman, S. Schmidt, S. Zadtootaghaj, M. G. Martini, and S. Möller, “An evaluation of video quality assessment metrics for passive gaming video streaming,” in *Proceedings of the 23rd packet video workshop*, 2018, pp. 7–12. [25](#)
- [98] Z. Luo, Y. Huang, X. Wang, R. Xie, and L. Song, “Vmaf oriented perceptual optimization for video coding,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5. [25](#)
- [99] C. G. Bampis, Z. Li, and A. C. Bovik, “Enhancing temporal quality measurements in a globally deployed streaming video quality predictor,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 614–618. [25](#)
- [100] K. Sampath, P. Venkatesan, P. Ramachandran, and K. Goswami, “Block-based temporal metric for video quality assessment,” in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, pp. 1–4. [25](#)

REFERENCES

- [101] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Traces and emergence of nonlinear programming*. Springer, 2014, pp. 247–258. [25](#), [26](#)
- [102] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 144–152. [25](#)
- [103] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, “Support vector regression machines,” in *Advances in neural information processing systems*, 1997, pp. 155–161. [25](#)
- [104] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, “A practical guide to support vector classification,” 2003. [26](#)
- [105] H. Q. Minh, P. Niyogi, and Y. Yao, “Mercer’s theorem, feature maps, and smoothing,” in *International Conference on Computational Learning Theory*. Springer, 2006, pp. 154–168. [27](#)
- [106] P. J. Burt and E. H. Adelson, “A multiresolution spline with application to image mosaics,” *ACM Transactions on Graphics (TOG)*, vol. 2, no. 4, pp. 217–236, 1983. [28](#)
- [107] P. Fuchs, *Virtual reality headsets-a theoretical and pragmatic approach*. CRC Press, 2017. [29](#)
- [108] Oculus developers, “Oculus Device Specifications,” <https://developer.oculus.com/design/oculus-device-specs/>, online; accessed 02-Jul-2020. [29](#)
- [109] The 360 Guy, “The Ultimate VR Headset Comparison Table: Every VR Headset Compared,” <http://www.threesixtycameras.com/vr-headset-comparison-table/>, Apr 2020, online; accessed 02-Jul-2020. [29](#)
- [110] I. Bauermann, M. Mielke, and E. Steinbach, *H.264 BASED CODING OF OMNIDIRECTIONAL VIDEO*. Dordrecht: Springer Netherlands, 2006, pp. 209–215. [Online]. Available: https://doi.org/10.1007/1-4020-4179-9_30 [30](#), [32](#)

-
- [111] T. El-Ganainy and M. Hefeeda, “Streaming virtual reality content,” *arXiv preprint arXiv:1612.08350*, 2016. [30](#), [31](#), [38](#), [39](#)
- [112] D. Liu, P. An, R. Ma, W. Zhan, and L. Ai, “Scalable omnidirectional video coding for real-time virtual reality applications,” *IEEE Access*, vol. 6, pp. 56 323–56 332, 2018. [30](#), [31](#)
- [113] G. Van der Auwera, M. Coban, and M. Karczewicz, “Sphere equator projection for efficient compression of 360-degree video,” Sep. 27 2018, uS Patent App. 15/926,732. [31](#)
- [114] M. Chochlík, “Scalable multi-GPU cloud raytracing with OpenGL,” in *The 10th International Conference on Digital Technologies 2014*. IEEE, 2014, pp. 87–95. [32](#)
- [115] M. Zhou, “AHG8: A study on compression efficiency of cube projection,” *Document JVET-D0022, Chengdu, CN*, 2016. [32](#)
- [116] Facebook developers, “Under the hood: Building 360 video,” <https://engineering.fb.com/video-engineering/under-the-hood-building-360-video/>, Oct 2015, online; accessed 02-Jul-2020. [34](#)
- [117] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “Viewport-Adaptive Encoding and Streaming of 360-Degree Video for Virtual Reality Applications,” in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 583–586. [34](#)
- [118] C. Brown, “Bringing pixels front and center in VR video,” *Retrieved March*, vol. 19, p. 2019, 2017. [34](#), [35](#)
- [119] C.-W. Fu, L. Wan, T.-T. Wong, and C.-S. Leung, “The rhombic dodecahedron map: An efficient scheme for encoding panoramic video,” *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 634–644, 2009. [35](#)
- [120] S. Akula, A. Singh, A. Dsouza, R. Gadde *et al.*, “AHG8: Efficient Frame Packing for Icosahedral Projection,” *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-E0029*, 2017. [35](#)

REFERENCES

- [121] H. Lin, C. Huang, C. Li, Y. Lee, J. Lin, and S. Chang, “AHG8: An Improvement on the Compact OHP Layout, document JVET-E0056, Joint Video Exploration Team of ITU-TSG16 WP3 and ISO,” IEC JTC1/SC29/WG11, Geneva, Switzerland, Tech. Rep., 2017. [35](#)
- [122] H. Lin, C. Li, J. Lin, S. Chang, and C. Ju, “AHG8: An Efficient Compact Layout for Octahedron Format, document JVETD0142, Joint Video Exploration Team of ITU-T SG16 WP3 and ISO,” IEC JTC1/SC29/WG11, Chengdu, China, Tech. Rep., 2016. [35](#)
- [123] M. Coban, G. Van der Auwera, and M. Karczewicz, “AHG8: Adjusted cubemap projection for 360-degree video,” *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC*, vol. 29, 2017. [35](#)
- [124] M. Yu, H. Lakshman, and B. Girod, “Content adaptive representations of omnidirectional videos for cinematic virtual reality,” in *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, 2015, pp. 1–6. [36](#)
- [125] J. Li, Z. Wen, S. Li, Y. Zhao, B. Guo, and J. Wen, “Novel tile segmentation scheme for omnidirectional video,” in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 370–374. [36](#)
- [126] C. Zhang, Y. Lu, J. Li, and Z. Wen, “AHG8: Segmented Sphere Projection for 360-degree video,” *Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-E0025*, 2017. [36](#)
- [127] Facebook developers, “End-to-end optimizations for dynamic streaming,” <https://engineering.fb.com/video-engineering/end-to-end-optimizations-for-dynamic-streaming/>, Feb 2017, online; accessed 02-Jul-2020. [39](#)
- [128] O. Niamut, M. Prins, R. v. Brandenburg, and A. Havekes, “Spatial tiling and streaming in an immersive media delivery network,” 2011. [40](#)
- [129] Y. Sanchez, R. Skupin, and T. Schierl, “Compressed domain video processing for tile based panoramic streaming using HEVC,” in *2015 IEEE International Conference on Image Processing (ICIP)*, Sept 2015, pp. 2244–2248. [41](#), [58](#)

-
- [130] ISO/IEC, “ISO/IEC 23009-1: 2014: Information technology-Dynamic adaptive streaming over HTTP (DASH)-Part 1: Media presentation description and segment formats,” 2014. [41](#)
- [131] L. De Cicco, S. Mascolo, V. Palmisano, and G. Ribezzo, “Reducing the network bandwidth requirements for 360 immersive video streaming,” *Internet Technology Letters*, vol. 2, no. 4, p. e118, 2019. [42](#), [57](#), [81](#), [82](#)
- [132] O. A. Niamut, E. Thomas, L. D’Acunto, C. Concolato, F. Denoual, and S. Y. Lim, “Mpeg dash srd: Spatial relationship description,” in *Proc. of the 7th International Conference on Multimedia Systems*, ser. MMSys ’16. New York, NY, USA: ACM, 2016, pp. 5:1–5:8. [42](#), [95](#)
- [133] Y. Wu, G. J. Sullivan, and Y. Zhang, “Control data for motion-constrained tile set,” Aug. 29 2017, uS Patent 9,749,627. [45](#)
- [134] R. Skupin, Y. Sanchez, K. Suehring, T. Schierl, E. Ryu, and J. Son, “Temporal mcts coding constraints implementation,” in *120th meeting of ISO/IEC JTC1/SC29/WG11, MPEG*, vol. 120, 2017, p. m41626. [45](#)
- [135] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello, “360 innovations for panoramic video streaming,” in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, 2017, pp. 50–56. [45](#), [54](#)
- [136] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, “An http/2-based adaptive streaming framework for 360 virtual reality videos,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 306–314. [45](#)
- [137] D. V. Nguyen, T. T. Le, S. Lee, and E.-S. Ryu, “Shvc tile-based 360-degree video streaming for mobile vr: Pc offloading over mmwave,” *Sensors*, vol. 18, no. 11, p. 3728, 2018. [45](#)
- [138] C. Ozcinar, A. De Abreu, and A. Smolic, “Viewport-aware adaptive 360 video streaming using tiles for virtual reality,” in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2174–2178. [45](#)

REFERENCES

- [139] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “HEVC-compliant tile-based streaming of panoramic video for virtual reality applications,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 601–605. [45](#)
- [140] B. Choi, Y. Wang, M. Hannuksela, Y. Lim, and A. Murtaza, “Information technology-coded representation of immersive media (MPEG-I)–part 2: Omni-directional media format,” *ISO/IEC*, pp. 23 090–2, 2017. [45](#), [46](#), [80](#)
- [141] M. M. Hannuksela, Y.-K. Wang, and A. Hourunranta, “An overview of the OMAF standard for 360 video,” in *2019 Data Compression Conference (DCC)*. IEEE, 2019, pp. 418–427. [47](#), [50](#)
- [142] D. Singer, M. Z. Visharam, Y. Wang, and T. Rathgen, “ISO/IEC 14496-15: 2004/Amd2: SVC File Format,” *International Standardization Organization*, 2007. [49](#)
- [143] G. Cofano, L. De Cicco, and S. Mascolo, “Characterizing adaptive video streaming control systems,” in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2729–2734. [50](#)
- [144] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, “What happens when http adaptive streaming players compete for bandwidth?” in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*, 2012, pp. 9–14. [52](#)
- [145] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, “Confused, timid, and unstable: picking a video streaming rate is hard,” in *Proceedings of the 2012 internet measurement conference*, 2012, pp. 225–238. [52](#)
- [146] T. Kupka, P. Halvorsen, and C. Griwodz, “Performance of on-off traffic stemming from live adaptive segmented http video streaming,” in *37th Annual IEEE Conference on Local Computer Networks*. IEEE, 2012, pp. 401–409. [52](#)
- [147] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with pensieve,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 197–210. [53](#)

-
- [148] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, “Towards network-wide qoe fairness using openflow-assisted adaptive video streaming,” in *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*, 2013, pp. 15–20. [53](#)
- [149] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, “Design and experimental evaluation of network-assisted strategies for http adaptive streaming,” in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016, pp. 1–12. [54](#)
- [150] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, “Optimizing 360 video delivery over cellular networks,” in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, 2016, pp. 1–6. [54](#)
- [151] Y. Hu, Y. Liu, and Y. Wang, “Vas360: Qoe-driven viewport adaptive streaming for 360 video,” in *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2019, pp. 324–329. [54](#)
- [152] A. Mavlankar and B. Girod, “Video streaming with interactive pan/tilt/zoom,” in *High-Quality Visual Experience*. Springer, 2010, pp. 431–455. [54](#)
- [153] S. Petrangeli, G. Simon, and V. Swaminathan, “Trajectory-based viewport prediction for 360-degree virtual reality videos,” in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2018, pp. 157–160. [54](#)
- [154] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, “Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming,” in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6. [54](#)
- [155] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, “Shooting a moving target: Motion-prediction-based transmission for 360-degree videos,” in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170. [54](#)

REFERENCES

- [156] M. Jamali, S. Coulombe, A. Vakili, and C. Vazquez, “Lstm-based viewpoint prediction for multi-quality tiled video coding in virtual reality streaming,” in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5. [54](#)
- [157] F. Duanmu, E. Kurdoglu, S. A. Hosseini, Y. Liu, and Y. Wang, “Prioritized buffer control in two-tier 360 video streaming,” in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 13–18. [54](#)
- [158] J. Yu and Y. Liu, “Field-of-view prediction in 360-degree videos with attention-based neural encoder-decoder networks,” in *Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems*, 2019, pp. 37–42. [54](#)
- [159] A. Nguyen, Z. Yan, and K. Nahrstedt, “Your attention is unique: Detecting 360-degree video saliency in head-mounted display for head movement prediction,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1190–1198. [55](#)
- [160] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, “Predicting head movement in panoramic video: A deep reinforcement learning approach,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2693–2708, 2018. [55](#)
- [161] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, “Gaze prediction in dynamic 360 immersive videos,” in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5333–5342. [55](#)
- [162] Q. Yang, J. Zou, K. Tang, C. Li, and H. Xiong, “Single and sequential viewports prediction for 360-degree video streaming,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5. [55](#)
- [163] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, “Fixation prediction for 360 video streaming in head-mounted virtual reality,” in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2017, pp. 67–72. [55](#)

-
- [164] C. Ozcinar, J. Cabrera, and A. Smolic, “Visual attention-aware omnidirectional video streaming using optimal tiles for virtual reality,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 217–230, 2019. [55](#)
- [165] G. Ribezzo, G. Samela, V. Palmisano, L. De Cicco, and S. Mascolo, “Reducing the network bandwidth requirements for immersive video streaming,” in *Proc. of Second International Balkan Conference on Communications and Networking*. Podgorica: IEEEComSoc, June 2018. [57](#)
- [166] C. Concolato, J. L. Feuvre, F. Denoual, E. Nassor, N. Ouedraogo, and J. Taquet, “Adaptive streaming of hevc tiled videos using mpeg-dash,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2017. [58](#), [61](#), [81](#)
- [167] M. Hosseini and V. Swaminathan, “Adaptive 360 vr video streaming: Divide and conquer,” in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 107–110. [58](#), [80](#)
- [168] R. Skupin, Y. Sanchez, Y.-K. Wang, M. M. Hannuksela, J. Boyce, and M. Wien, “Standardization status of 360 degree video coding and delivery,” in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4. [58](#)
- [169] T. Biedert, P. Messmer, T. Fogal, and C. Garth, “Hardware-accelerated multi-tile streaming for realtime remote visualization.” in *EGPGV*, 2018, pp. 33–43. [58](#)
- [170] FFmpeg developers, “lavc/vaapi_encode_h265: add h265 tile encoding support,” <https://github.com/FFmpeg/FFmpeg/commit/43a08d907ba765677254b4816f246a8ecfd7ff78>, Jul 2020, online; accessed 20-Jul-2020. [58](#)
- [171] NVIDIA developers, “HEVC tile support? NVENC nvidia SDK,” <https://forums.developer.nvidia.com/t/hevc-tile-support-nvenc-nvidia-sdk/69469>, Jul 2019, online; accessed 20-Jul-2020. [58](#)
- [172] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Hämmäläinen, “Kvazaar: Open-source hevc/h.265 encoder,” in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016. [58](#), [84](#)

REFERENCES

- [173] A. Zare, A. Aminlou, and M. M. Hannuksela, “6K Effective Resolution with 4K HEVC Decoding Capability for OMAF-compliant 360-degree Video Streaming,” in *Proc. of the 23rd Packet Video Workshop*, ser. PV '18, 2018, pp. 72–77. [58](#)
- [174] H. Hristova, X. Corbillon, G. Simon, V. Swaminathan, and A. Devlic, “Heterogeneous spatial quality for omnidirectional video,” in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, Aug 2018, pp. 1–6. [58](#), [59](#), [61](#), [81](#)
- [175] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, “Viewport-adaptive navigable 360-degree video delivery,” in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7. [61](#)
- [176] R. G. Youvalari, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, “Efficient coding of 360-degree pseudo-cylindrical panoramic video for virtual reality applications,” in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 525–528. [61](#)
- [177] G. Ribezzo, G. Samela, V. Palmisano, L. De Cicco, and S. Mascolo, “A dash video streaming system for immersive contents,” in *Proc. of ACM Multimedia Systems Conference*. Amsterdam: ACM, June 2018, pp. 525–528. [67](#), [84](#), [99](#)
- [178] G. Ribezzo, L. De Cicco, V. Palmisano, and S. Mascolo, “A DASH 360 immersive video streaming control system,” *Internet Technology Letters*, 2019. [67](#)
- [179] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan, “Mahimahi: Accurate Record-and-Replay for {HTTP},” in *2015 {USENIX} Annual Technical Conference ({USENIX}{ATC} 15)*, 2015, pp. 417–429. [73](#)
- [180] X. Corbillon, F. De Simone, and G. Simon, “360-degree video head movement dataset,” in *Proc. of ACM Multimedia Systems Conference*, 2017, pp. 199–204. [73](#), [94](#), [100](#), [101](#)
- [181] M. Jamali, F. Golaghazadeh, S. Coulombe, A. Vakili, and C. Vazquez, “Comparison of 3d 360-degree video compression performance using different projections,” in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 2019, pp. 1–6. [80](#)

-
- [182] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, “Rubiks: Practical 360-degree streaming for smartphones,” in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 482–494. [81](#)
- [183] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, “Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114. [81](#)
- [184] S. Rossi, C. Ozcinar, A. Smolic, and L. Toni, “Do Users Behave Similarly in VR? Investigation of the User Influence on the System Design,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2, pp. 1–26, 2020. [81](#), [82](#), [100](#), [101](#)
- [185] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, “Toward a practical perceptual video quality metric,” *The Netflix Tech Blog*, vol. 6, 2016. [83](#), [88](#)
- [186] M. Orduna, C. Diaz, L. Munoz, P. Perez, I. Benito, and N. Garcia, “Video multimethod assessment fusion (vmaf) on 360vr contents,” *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 22–31, 2020. [83](#)
- [187] G. Ribezzo, L. De Cicco, V. Palmisano, and S. Mascolo, “Tapas-360: A tool for the design and experimental evaluation of 360 video streaming systems,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4477–4480. [93](#)
- [188] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, “Tapas: a tool for rapid prototyping of adaptive streaming algorithms,” in *Proc. Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, 2014, pp. 1–6. [94](#), [95](#), [100](#)
- [189] A. Zabrovskiy, E. Kuzmin, E. Petrov, C. Timmerer, and C. Mueller, “Advise: Adaptive video streaming evaluation framework for the automated testing of media players,” in *Proc. ACM MMSys ’17*, 2017, pp. 217–220. [94](#)

REFERENCES

- [190] Y. Rai, J. Gutiérrez, and P. Le Callet, “A dataset of head and eye movements for 360 degree images,” in *Proc. of ACM Multimedia Systems Conference*, 2017, pp. 205–210. [94](#), [99](#), [100](#), [101](#)



**Relazione sull'attività complessivamente svolta dal dottorando nei III anni di
Corso di Dottorato in Ingegneria Elettrica e dell'Informazione (ciclo XXXIII)**

Il sottoscritto dott. Giuseppe Ribezzo, nato a Francavilla Fontana il 17/02/1989, ha conseguito la laurea in Ingegneria delle Telecomunicazioni presso il Politecnico di Bari in data 24/03/2016, con la votazione di 110/110 e lode discutendo la tesi dal titolo "**Sviluppo di Architetture di Telecomunicazioni Data-Centriche per Sistemi di Trasporto Intelligente**", Relatore: **Prof. Grieco Luigi Alfredo**.

Il programma di ricerca individuale proposto/assegnato dal Collegio dei Docenti svolto ha per titolo: "**Algoritmi di controllo per lo streaming video adattivo in applicazioni in Realtà Virtuale e Aumentata**" con relatore il **Prof. Ing. Mascolo Saverio**.

Nello specifico, l'attività di ricerca del dottorando è stata finalizzata a studiare e modellare metodologie per la riduzione del consumo di banda in applicazioni di streaming di video immersivi su rete Internet.

Attività didattiche relative al I anno

Nel I anno del corso di dottorato in epigrafe il sottoscritto Giuseppe Ribezzo, ha frequentato i seguenti percorsi formativi organizzati dal Corso di dottorato o offerti da Università /Enti di ricerca (con indicazione dei CFU eventualmente conseguiti) :

- **How to write a technical paper and to present it effectively to an educated audience** (3 CFU, con esame)
- **Middleware and architecture for industry 4.0** (3 CFU, con esame)
- **Introduction to statistical mechanics and applications** (1.5 CFU, solo frequenza)
- **Theory and Applications of stochastic processes** (1.5 CFU, solo frequenza)

Il sottoscritto ha altresì partecipato ai seguenti seminari e convegni scientifici su tematiche afferenti al dottorato in oggetto:

- **BalkanCom 2018, Second International Balkan Conference on Communications and Networking**, tenutasi in Podgorica, Montenegro, 6 - 8 giugno, 2018. Nella presente conferenza è stato presentato l'articolo in 2.

Relazione attività di ricerca

- **MMSys2018, ACM Multimedia Systems Conference**, tenutasi in Amsterdam, Netherlands, 12 - 15 giugno, 2018. Nella presente conferenza è stata presentata una demo del prototipo di piattaforma di video streaming esposta nell'articolo in 1.
- **S.I.D.R.A. (Società Italiana Docenti e Ricercatori in Automatica) PhD Summer School**, tenutasi in Bertinoro (FC), 9-14 luglio, 2018, avente due tematiche:
 - **Adaptive Control Systems: Methodologies for Analysis and Synthesis**, 15 ore;
 - **Optimization Methods for Decision Making over Networks**, 15 ore, con esame finale.
- **Machine Learning and Condition-based Monitoring**, tenutosi in Bari (BA), 17 luglio, 2018, della durata di 2 ore.
- **Beyond the Desktop Metaphor: Opportunities and Challenges of Creating Effective Augmented Reality User Experiences**, tenutosi in Bari (BA), 18 luglio, 2018, della durata di 2 ore.
- **Data science and its application with deep learning artificial neural network**, tenutosi in Bari (BA), 24 luglio, 2018, della durata di 2 ore.

Nel corso del I anno di dottorato, il sottoscritto ha anche partecipato alle attività connesse ad alcuni progetti di ricerca. In particolare:

- Progetto "HORIZON 2020" PON I&C 2014-2020 "**CLIPS: a Cloud-based platform for Immersive adaptive video Streaming**" (coordinatore Prof. Mascolo Saverio).

Inoltre, il sottoscritto ha altresì partecipato ai seguenti corsi di formazione su tematiche afferenti al dottorato in oggetto:

- **Corso Amministratore di Reti – Cisco CCENT/CCNA**

ottentendo la certificazione *Cisco Academy CCNA Routing and Switching: Scaling Networks*.

Attività didattiche relative al II anno

Nel II anno del corso di dottorato in epigrafe il sottoscritto Giuseppe Ribezzo, ha frequentato i seguenti percorsi formativi organizzati dal Corso di dottorato o offerti da Università /Enti di ricerca (con indicazione dei CFU eventualmente conseguiti) :

- **Elements of Probability for Engineering Sciences** (3 CFU, frequenza + conseguimento esame)
- **Theory and Applications of stochastic processes** (1.5 CFU, solo conseguimento esame)

Il sottoscritto ha altresì partecipato ai seguenti seminari e convegni scientifici su tematiche afferenti al dottorato in oggetto:

- **Data-driven Modeling and Optimization: a Networking Perspective**, tenutosi in Bari (BA), 13 giugno, 2019, della durata di 2 ore.
- **National Instruments Workshop**, tenutosi in Bari (BA), 09-10 aprile, 2019, della durata di 2 giorni.

Relazione attività di ricerca

- **Networking Research Topics: Past, Present and Future inspired by Mario Gerla**, tenutosi in Milano (MI), 03 giugno, 2019, della durata di 1 giorno.
- **Summer School of Information Engineering (SSIE Summer School) 2019**: tenutosi in Bressanone (BZ), 08-12 luglio, 2019, della durata di 30 ore.

Nel corso del II anno di dottorato, il sottoscritto ha anche partecipato alle attività connesse ad alcuni progetti di ricerca. In particolare:

- Progetto "HORIZON 2020" PON I&C 2014-2020 "**CLIPS: a Cloud-based platform for Immersive adaPtive video Streaming**" (coordinatore Prof. Mascolo Saverio).

Inoltre, il sottoscritto ha altresì partecipato ai seguenti corsi di formazione su tematiche afferenti al dottorato in oggetto:

- *Corso Amministratore di Reti – Cisco CCENT/CCNA*

ottendendo la certificazione *Cisco Academy CCNA Routing and Switching: Connecting Networks*.

Attività didattiche relative al III anno

Nel III anno del corso di dottorato in epigrafe il sottoscritto Giuseppe Ribezzo, sta attualmente frequentando i seguenti percorsi formativi organizzati dal Corso di dottorato o offerti da Università /Enti di ricerca (con indicazione dei CFU eventualmente conseguiti) :

- **Software-based methods for modern control systems design** (solo frequenza, 1.5 CFU)
- **Emerging methodologies and technologies for the Cyber Security** (solo frequenza, 1.5 CFU)

Il sottoscritto ha altresì partecipato ai seguenti seminari e convegni scientifici su tematiche afferenti al dottorato in oggetto:

- **International Workshop on Smart Mobility in Future Cities: The Apulia Industry Summit**, tenutosi in Bari (BA), 06 ottobre, 2019, della durata di un giorno.
- **Mathematics for Engineering Applications**: tenutosi in Bari (BA), 27-31 gennaio, 2020, della durata di 30 ore.
- **ACM Multimedia 2020, 28th ACM International Conference on Multimedia**, tenutosi in Seattle, United States, 12-16 Ottobre, 2020. Nella presente conferenza è stato presentato l'articolo in 2.

Nel corso del III anno di dottorato, il sottoscritto sta attualmente partecipando alle attività connesse ad alcuni progetti di ricerca. In particolare:

- Progetto "HORIZON 2020" PON I&C 2014-2020 "**CLIPS: a Cloud-based platform for Immersive adaPtive video Streaming**" (coordinatore Prof. Mascolo Saverio).

Descrizione sintetica dell'attività di ricerca

Attività di ricerca I anno

L'attività di ricerca svolta nel primo anno è stata suddivisa in due fasi.

Nella prima fase è stato studiato lo stato dell'arte riguardante sia le proiezioni ed i formati maggiormente utilizzati per la *mappatura sphere-to-plane* di immagini e riprese video a 360°, sia le tecniche di *quality adaptation* proposte in letteratura. In primo luogo, sono state identificate e comparate due diverse tipologie di mappature:

1. mappature a **qualità fissa**;
2. mappature a **qualità variabile**;

Ogni tipologia di mappatura è stata analizzata sulla base della qualità complessiva del frame video risultante, della complessità dell'algoritmo utilizzato per generare il frame e dell'occupazione di storage, in maniera tale da trovare il giusto compromesso tra i parametri prima citati.

In secondo luogo, diverse strategie per implementare la *quality adaptation* sono state considerate. Per **quality adaptation** si intendono quelle tecniche che consentono di mantenere a qualità originale solo quelle porzioni di video che sono di reale interesse per l'utente - denominata **Region Of Interest (ROI)** - e ridurre la qualità nelle altre porzioni. In letteratura, sono state studiate diverse tecniche di produzione di contenuti video a 360° che implementano questo approccio, i quali possono essere catalogati in:

- **real-time**, nella quale selezione della ROI, codifica, consegna e riproduzione del video a 360° vengono effettuati in real-time per ogni utente;
- **off-line**, nella quale alcune assunzioni sono fatte sulla statistica di visualizzazione degli utenti consentendo che la selezione della ROI e la codifica del video a 360° possa avvenire precedentemente alla consegna e riproduzione dello stesso.

Questa tipologia di strategie consente di ottenere un considerevole risparmio in termini di consumo di banda, ma necessita di un opportuno design sia del meccanismo di produzione del contenuto multimediale, sia dell'algoritmo di controllo che gestisca lo *switch* tra le versioni del video con differenti livelli di qualità in maniera tale da garantire una *Quality of Experience* che sia la massima possibile per l'utente.

Relazione attività di ricerca

In parallelo a queste attività, all'interno del progetto PON I&C 2014-2020 CLIPS si è tenuta una ampia discussione finalizzata a identificare tutte le caratteristiche e le funzionalità di una piattaforma per lo streaming di video immersivi. Inoltre, sono stati vagliati i casi d'uso, e le suddette caratteristiche e funzionalità sono state adattate a ciascuno degli scenari considerati.

La seconda parte del lavoro ha visto l'introduzione di una tecnica di *quality adaptation off-line* innovativa concepita per la produzione di contenuti video a 360°, di cui si vuole misurare l'efficacia nel ridurre il consumo di banda in rapporto allo stato dell'arte senza degradare la Quality Of Experience per l'utente.

La tecnica ideata è indipendente dalla tipologia di *encoder* utilizzato e sfrutta algoritmi e tecnologie maturi, perciò risulta essere impiegabile sulla maggiorparte delle piattaforme di video streaming e dispositivi mobili oggi in commercio.

In particolare, per implementare il concetto di ROI, essa fa uso delle funzionalità di *slicing* e *rescaling*, consentendo di suddividere un video in fette (*slice* in inglese) e di riscaldare le *slices* che non appartengono alla ROI considerata, ottenendo in tal modo la riduzione nel consumo di banda. Si noti che le funzionalità di *slicing* e *rescaling* sono supportate nativamente dagli *encoder-decoder* esistenti e consentono di sfruttare l'accelerazione hardware delle schede grafiche. L'algoritmo risulta essere in tal modo energeticamente efficiente e quindi adatto ad essere utilizzato su dispositivi mobili. Complessivamente, vengono selezionate un numero di ROI tale da coprire l'intera area del video. Per ciascuna ROI, viene prodotta una clip video, denominata *view*, che comprende la ROI a risoluzione nativa mentre la restante parte del video viene riscaldata di un opportuno *scaling factor*. Lato client, il player scarica la *view* più opportuna a seconda della ROI visualizzata dall'utente. Si noti che il player necessita di un algoritmo di controllo per la *view-selection* che da un lato massimizzi la qualità visiva percepita dall'utente e dall'altro garantisca l'assenza di eventi di rebuffering.

La tecnica di *content generation* proposta è stata valutata attraverso un'estensiva campagna di test, i risultati dei quali sono disponibili nelle pubblicazioni 1. e 2. sotto citati.

La prossima fase del lavoro di ricerca prevede l'ampliamento del lavoro svolto fino ad ora da più punti di vista. In primo luogo, saranno studiati gli algoritmi e le tecnologie per la creazione di contenuti olografici, e tali contenuti saranno integrati nella piattaforma di video streaming. In secondo luogo, saranno studiati e valutati gli algoritmi di controllo per il

supporto della *view-selection*, in maniera tale da trovare il giusto trade-off tra probabilità di eventi di rebuffering e qualità visiva.

Attività di ricerca II anno

L'attività di ricerca svolta nel secondo anno è stata suddivisa in due fasi.

Nella prima fase è stato studiato lo stato dell'arte riguardante gli algoritmi di controllo e le metodologie di design architetturale proposte nella letteratura scientifica per l'*adaptive streaming* dei video 2D. È stato rilevato che il compito dell'algoritmo di controllo in una piattaforma di *adaptive streaming* è la massimizzazione della *Quality of Experience* (QoE) percepita dagli utenti. In tal modo, un algoritmo di controllo deve perseguire i seguenti obiettivi (in ordine decrescente di importanza):

1. evitare interruzioni nella riproduzione (eventi di rebuffering);
2. massimizzare la qualità del video (livello o bitrate);
3. minimizzare il tempo di avvio;
4. minimizzare il numero di switch di livello video.

Per soddisfare tali requisiti, l'approccio convenzionale prevede che siano usati congiuntamente due algoritmi:

- un algoritmo per selezionare dinamicamente il livello del video, che dovrebbe idealmente corrispondere alla larghezza di banda disponibile;
- un controller del playout buffer che viene utilizzato per assorbire le variazioni della larghezza di banda ed evitare interruzioni nella riproduzione.

In particolare, gli algoritmi di controllo del playout buffer progettati in letteratura seguono uno dei due seguenti approcci:

- Approccio *rate-based*, nel quale il buffer è controllato sulla base del rate o della banda di ricezione;
- Approccio *level-based*, nel quale il buffer è controllato sulla base del livello video ricevuto.

Risulta chiaro che un'appropriata caratterizzazione del *playout buffer* (cioè del buffer che sottende alla memorizzazione dei frame video che saranno riprodotti) assume grande rilevanza nello sviluppo di una piattaforma di video streaming efficiente.

Alla luce dello studio dello stato dell'arte, la seconda parte del lavoro ha previsto la preliminare progettazione di una architettura di controllo per la distribuzione di video

Relazione attività di ricerca

panoramici in modalità adattiva. Tale architettura di controllo è stata disegnata con un approccio modulare e consente il riutilizzo (con alcune modifiche di limitata invasività) degli algoritmi di selezione del bitrate progettati per video 2D. Inoltre, essa è stata disegnata in maniera tale da sfruttare l'algoritmo di *content generation* proposta durante il primo anno di questo dottorato.

Nello specifico, essa prevede l'utilizzo concorrente di due controllori:

- *Quality Selection Algorithm* (QSA), che consente di selezionare il livello video più appropriato;
- *View Selection Algorithm* (VSA), avente il compito di selezionare la view più adeguata per adattarsi dinamicamente al movimento della testa dell'utente durante la sessione di streaming;

Nello specifico, come QSA è stato scelto l'algoritmo ELASTIC, in quanto risulta garantire la migliore performance relativamente al numero ed alla frequenza degli eventi di rebuffering. Per quanto riguarda il controllore VSA, è stato implementato un algoritmo che, tenendo traccia del movimento della testa dell'utente, seleziona la view più opportuna sulla base del viewport correntemente visualizzato e della banda disponibile, in maniera tale da minimizzare il numero di switch e la probabilità di rebuffering. È da notare che il controllore VSA è un componente innovativo non presente nelle piattaforme di video streaming adattivo ed è stato progettato da zero.

L'introduzione del VSA ha apportato notevoli cambiamenti al comportamento dinamico del *playout buffer* atteso con il solo QSA. Pertanto, si è rivelata necessaria una nuova caratterizzazione del *playout buffer* che tenesse conto del lavoro combinato di entrambi i controllori.

Infine, è stata condotta una analisi sulla variazione delle performance ottenibili relativamente alle metriche di Rebuffering ratio (rapporto tra eventi di rebuffering e durata della sessione di streaming), bitrate medio e qualità media percepita dall'utente.

I risultati di tale campagna sperimentale sono stati pubblicati in 5.

In parallelo a queste attività, all'interno del progetto PON I&C 2014-2020 CLIPS si è tenuta una ampia discussione sulle modalità di interlavoro ed integrazione delle diverse componenti formanti una piattaforma per lo streaming di video immersivi. Inoltre, una versione prototipale di piattaforma è stata implementata.

Relazione attività di ricerca

La prossima fase del lavoro di ricerca prevede l'ampliamento del lavoro svolto fino ad ora da più punti di vista. Infatti, il sistema di controllo composto da QSA e VSA risulta essere pilotato da un insieme di parametri che richiede la risoluzione di un problema di ottimizzazione per la taratura ottimale. Futuro lavoro di ricerca sarà incentrato sullo studio di tali algoritmi di ottimizzazione. In secondo luogo, l'algoritmo VSA implementato fa uso di un algoritmo reattivo ai movimenti dell'utente. Futuro lavoro sarà quindi incentrato sui possibili utilizzi di algoritmi predittivi o basati su tecniche di machine learning.

Attività di ricerca III anno

L'attività di ricerca svolta nel terzo anno è suddivisa in due fasi.

Nella prima fase è stato studiato lo stato dell'arte riguardante gli algoritmi per l'ottimizzazione della QoE di video immersivi all'utente.

In sintesi, sono state rilevate due strategie di ottimizzazione, che possono essere descritte come:

- *content-aware*, tese ad ottimizzare la QoE di video a 360 sulla base del contenuto visivo della scena rappresentata;
- *viewport-adaptive*, dove l'ottimizzazione avviene sulla base del contenuto correntemente visualizzato dall'utente.

Per quanto riguarda gli algoritmi *content-aware*, l'ottimizzazione del contenuto scenico avviene nella maggioranza dei casi in post-produzione; d'altra parte, gli algoritmi *viewport-adaptive* richiedono che l'esecuzione dell'algoritmo di ottimizzazione avvenga quasi in real-time, cioè contestualmente alla fase di visualizzazione del contenuto o nell'immediata fase precedente la visualizzazione.

Risulta chiara la possibilità di utilizzare le diverse implementazioni di entrambi gli algoritmi in cascata.

Inoltre, sono stati identificati i due approcci maggiormente utilizzati in letteratura scientifica per la compressione dei contenuti video 360, denominati:

- *Variable resolution (VRES)*, che prevede la creazione di contenuti video 360 nel quale una particolare regione spaziale, denominata *Region of Interest (ROI)*, viene codificata alla risoluzione nativa mentre le regioni al di fuori della ROI considerata sono codificate ad una risoluzione inferiore. il risultato è una qualità visiva per la ROI

Relazione attività di ricerca

considerata pari al video originale, ma con una richiesta di bitrate per la codifica complessivamente inferiore;

- *Variable quantization (VQP)*, dove la riduzione della qualità visiva al di fuori della ROI considerata è ottenuta mediante la riduzione del parametro di quantizzazione utilizzato dall'encoder.

Alla luce dello studio dello stato dell'arte, nella seconda parte del lavoro è stata condotta un'analisi comparativa volta a stabilire le performance ottenibili relativamente alle due tecniche di compressione dei contenuti video 360° sopra citate, ovvero VRES e VQP, attraverso una massiccia campagna di test sperimentali. Come parametri di performance sono state scelte le metriche di bitrate medio e qualità media percepita dall'utente.

I risultati di tale campagna sperimentale saranno pubblicati in futuro.

Inoltre, è stata implementata una piattaforma di distribuzione di contenuti immersivi DASH-compliant, perfezionando il dimostratore progettato in 2. ed integrando l'algoritmo di content generation sviluppato in 1.. La piattaforma di distribuzione è stata sottoposta ad una intensa campagna sperimentale i cui risultati sono stati divulgati in 6.

Parallelamente, mi sono occupato del design e dello sviluppo di uno strumento, denominato TAPAS-360 che estende il tool open-source TAPAS [1]. Lo strumento è stato progettato per permettere sia la prototipazione rapida degli algoritmi di viewport adaptivity proposti per la distribuzione di contenuti panoramici e facilitare il compito di confrontare sperimentalmente diversi algoritmi.

TAPAS-360 è stato progettato con un approccio modulare che permette al ricercatore di concentrarsi solo sullo sviluppo dell'algoritmo che coinvolge un particolare componente con un limitato intervento sul codice degli altri componenti.

Inoltre, lo strumento è stato progettato per diminuire il carico di calcolo richiesto per ogni flusso video, disabilitando il processo di decodifica senza interferire con la dinamica della sessione di streaming, ad esempio la dinamica del buffer di playout. Di conseguenza, è possibile effettuare una sperimentazione massiccia che coinvolge un gran numero di flussi contemporanei sulla stessa macchina.

Relazione attività di ricerca

Infine, lo strumento può essere facilmente esteso con comuni strumenti di emulazione di rete (come, ad esempio, MahiMahi [2]) per eseguire sperimentazioni in un ambiente di rete controllato.

In questo modo la riproducibilità dei risultati ottenuti viene implementata in modo immediato.

I risultati di tale campagna sperimentale sono stati pubblicati in 7.

In parallelo a queste attività, all'interno del progetto PON I&C 2014-2020 CLIPS si tiene una ampia discussione sulle modalità di interlavoro ed integrazione delle diverse componenti formanti una piattaforma per lo streaming di video immersivi. Inoltre, si stanno attualmente tenendo diverse campagne sperimentali che rileveranno gli indicatori chiave delle performance raggiungibili dalla versione prototipale della piattaforma progettata.

In breve, i possibili sviluppi di ricerca in questo ambito riguardano:

- l'analisi e lo sviluppo di algoritmi per la compressione dei contenuti 360 con approccio GPU-friendly;
- l'analisi delle tecniche di compressione di video 360 in formato volumetrico;
- identificazione di ulteriori use cases delle tecnologie di Computer Vision nell'ambito dell'industria 4.0.

Nell'ambito delle attività di ricerca sono state prodotte le seguenti pubblicazioni scientifiche:

1. Ribezzo, G., Samela, G., Palmisano, V., De Cicco, L., & Mascolo, S. (2018, June). *A DASH video streaming system for immersive contents*. In Proceedings of the 9th ACM Multimedia Systems Conference (pp. 525-528). ACM.
2. Ribezzo, G., Palmisano, V., De Cicco, L., & Mascolo, S. (2018, June). *A DASH video streaming system for immersive contents*. In Proceedings of the 2nd International Balkan Conference on Communications and Networking, June 6-8, 2018. IEEE Communication Society.
3. Losciale, M., Boccadoro, P., Piro, G., Ribezzo, G., Grieco, L. A., & Blefari-Melazzi, N. (2018, May). *A novel ICN-based communication bus for Intelligent Transportation Systems*. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.
4. Vogli, E., Ribezzo, G., Grieco, L. A., & Boggia, G. (2018). *Fast network joining algorithms in industrial IEEE 802.15. 4 deployments*. *Ad Hoc Networks*, 69, 65-75.

Relazione attività di ricerca

5. De Cicco, L., Mascolo, S., Palmisano, V., & Ribezzo, G. (2019). Reducing the network bandwidth requirements for 360° immersive video streaming. *Internet Technology Letters*, 2(4), e118.
6. Ribezzo, G., De Cicco, L., Palmisano, V., & Mascolo, S. A DASH 360° Immersive Video Streaming Control System. *Internet Technology Letters*.
7. Ribezzo, G., De Cicco, L., Palmisano, V., & Mascolo, S. TAPAS-360 A Tool for the Design and Experimental Evaluation of 360° Video Streaming Systems. In publishing at 28th ACM International Conference on Multimedia.

Il Tutor

Il Dottorando

MASCOLO SAVERIO



RIBEZZO GIUSEPPE



RIFERIMENTI

- [1] De Cicco, L., Caldaralo, V., Palmisano, V., & Mascolo, S. (2014, December). TAPAS: a Tool for rApid Prototyping of Adaptive Streaming algorithms. In *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming* (pp. 1-6).
- [2] Netravali, R., Sivaraman, A., Das, S., Goyal, A., Winstein, K., Mickens, J., & Balakrishnan, H. (2015). Mahimahi: Accurate record-and-replay for {HTTP}. In *2015 {USENIX} Annual Technical Conference ({USENIX}{ATC} 15)* (pp. 417-429).

I crediti sono stati calcolati facendo riferimento al Regolamento didattico della Scuola di Dottorato del Politecnico di Bari approvato dal Consiglio della SCUDO del 12.07.2018				
Didattica		Giorni	CFU	coeff [1CFU/1giorno]
Workshops	BalkanCom Conference	3	3	1
	MMSys2018 Conference	2	2	
	National Instruments	2	2	
	Networking Research Topics: Past, Present and Future inspired by Mario Gerla	1	2	
	International Workshop on Smart Mobility in Future Cities: The Apulia Industry Summit	1	1	
	ACM Multimedia 2020	5	5	
Presentazione Articoli	BalkanCom Conference		2	
	MMSys2018 Conference		2	
	ACM Multimedia 2020		2	
		Ore	CFU	coeff (1,5[CFU]/5 [ore])
Seminari	Data science and its application with deep learning artificial neural network	2	0,6	0,3
	Machine Learning and Condition-based Monitoring	2	0,6	
	Beyond the Desktop Metaphor: Opportunities and Challenges of Creating Effective Augmented Reality User Experiences	2	0,6	
	Data-driven Modeling and Optimization: a Networking Perspective	2	0,6	
Corsi	SIDRA Summer School 2018 Part II	18	4	
	SIDRA Summer School 2018 Part I	18	2	
	Networking Accademy CCNA 3	30	3	
	Networking Accademy CCNA 4	30	3	
	How to write a technical paper and to present it effectively to an educated audience	SCUDO	3	
	Introduction to statistical mechanics and applications	SCUDO	1,5	
	Middleware and architecture for industry 4.0	SCUDO	3	
	Elements of Probability for Engineering Sciences	SCUDO	3	
	Theory and Applications of stochastic processes	SCUDO	3	
	SSIE Summer School 2019	30	5	
	Software-based methods for modern control systems design	SCUDO	1,5	
	Emerging methodologies and technologies for the Cyber Security	SCUDO	1,5	
	Mathematics for Engineering Applications	30	6	
CFU con esame finale (Didattica)				39,5
Attività di Ricerca e studio individuale I anno			34	
Attività di Ricerca e studio individuale II anno			46	
Attività di Ricerca e studio individuale III anno			46	
CFU totali			188,9	

Il Dottorando

GIUSEPPE RIBEZZO



Il Tutor

SAVERIO MASCOLO

