Understanding user interactivity for the next-generation immersive communication: design, optimisation, and behavioural analysis

Silvia Rossi

A dissertation submitted to
University College London
for the degree of
Doctor of Philosophy.

Department of Electronic and Electrical Engineering
University College London
February 16, 2022

I, Silvia Rossi, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

"...Traguardi che sono partenze
ed un tramonto che è come un mattino."
LL

Abstract

Recent technological advances have opened the gate to a novel way to communicate remotely still feeling connected. In these immersive communications, humans are at the centre of virtual or augmented reality with a full sense of immersion and the possibility to interact with the new environment as well as other humans virtually present. These next generation communication systems hide a huge potential that can invest in major economic sectors. However, they also posed many new technical challenges, mainly due to the new role of the final user: from merely passive to fully active in requesting and interacting with the content. Thus, we need to go beyond the traditional quality of experience research and develop user centric solutions, in which the whole multimedia experience is tailored to the final interactive user. With this goal in mind, a better understanding of how people interact with immersive content is needed and it is the focus of this thesis.

In this thesis, we study the behaviour of interactive users in immersive experiences and its impact on the next-generation multimedia systems. The thesis covers a deep literature review on immersive services and user centric solutions, before developing three main research strands. First, we implement novel tools for behavioural analysis of users navigating in a 3-DoF Virtual Reality (VR) system. In detail, we study behavioural similarities among users by proposing a novel clustering algorithm. We also introduce information theoretic metrics for quantifying similarities for the same viewer across contents. As second direction, we show the impact and advantages of taking into account user behaviour in immersive systems. Specifically, we formulate optimal user centric solutions *i)* from a server-side perspective and *ii)* a navigation aware adaptation logic for VR streaming platforms. We conclude by exploiting the aforementioned behavioural studies towards a more interactive immersive technology: a 6-DoF VR. Overall in this thesis, experimental results based on real navigation trajectories show key advantages of understanding any hidden patterns of user interactivity to be eventually exploited in engineering user centric solutions for immersive systems.

Impact Statement

Immersive communication, which allows people to connect and feel present despite them being remote, has been recognised as one of the digital technologies that will rocket fuel our economy. This has been even more amplified by the recent COVID-19 outbreak, during which immersive reality has been identified among the key technologies helping businesses bouncing back from the pandemic. The revolutionary novelty of this technology is the possibility for users to interact with digital elements (*i.e.*, objects and/or surrounding environment), and to feel a sense of engagement and presence in a virtual space. This level of immersiveness and realism offered in immersive realities, however, comes with many new open challenges. Due to their interactivity, each of the user will live their own experience and next-generation immersive multimedia systems need to support such heterogeneity. Consequently, fundamentally new solutions are required to tailor the whole immersive experience to the final interactive users. In this context, my research is aimed at better understanding how people interact with the immersive content, advancing human-centric (tailored) solutions for next-generation immersive multimedia systems.

Over the past four years, this research has let to outcomes that have filled existing gaps both in 3- and 6-Degree of freedom (DoF) systems. Main outcomes have been the study of the behaviour of interactive users and, for the first time, the development of new behavioural analysis tools and methodologies, specifically built for immersive environments. At first, this research focused on studying behavioural similarities among users in a 3-DoF Virtual Reality (VR) system, proposing a new clustering algorithm. Metrics from information-theory were also introduced to quantify similarities across users experiencing the same content and across contents experienced by one single user. Identify similarities in the navigation is a step forward in modelling how users behave in virtual environments and is a key factor to better optimise experiences around the users. To show the impact and advantages of taking into account user behaviour in immersive systems, optimal user-centric solutions were also formulated as part of this thesis. Thanks to an exchange program founded by the UK Royal Society, in 2020 I undertook an intern-

ship for 6 months at the Centrum Wiskunde & Informatica (CWI), the Netherlands. This has allowed my research to move a step forward extending the aforementioned behavioural studies towards more interactive and immersive technologies, such as 6-DoF VR.

The research presented in the thesis has brought to 8 peer-reviewed publications, many of cross-disciplinary nature, which I have authored and which appeared at competitive and impactful venues. These outputs have given me high visibility within the multimedia community, leading to covering also technical roles in our SIGMM society and in the prestigious MMSys conference. The impact of the research reported in this thesis has been also recognised by an ERCIM fellowship, granted to me on November 2021, which will allow me to further extend my research direction toward broader goals.

Acknowledgements

First and foremost, I would like to express my endless gratitude and thanks to my supervisor, Dr. Laura Toni, who supported and guided me and my research over the past years. Her unique enthusiasm and passion for research have inspired and motivated me since day zero. The many opportunities she provided me, fruitful collaborations, participation but also to be included in the organisation of conferences have been invaluable for growing and being already part of the multimedia community. Being your (first!) PhD student is a real honour and, the memories of this journey together will always be with me. *Grazie, Laura!*
I would like to thank also my second supervisor, Prof. Izzat Darwazeh, for supporting my PhD and for being a constant guidance for me and Laura. My gratitude also to my examiners, Prof. Sally Day and Dr. Thomas Maugey, for their time to review my thesis, all their appreciation of my work and the constructive discussions.

I would like to immensely thank Prof. Pablo Cesar for believing in me and my research, not only for welcoming in his group at CWI but also for the opportunity to be information director of ACM SIGMM records. My gratitude also to Dr. Irene Viola for the very productive collaboration during my time in Amsterdam. Thank you both for your invaluable supervision and for all the social moments: my experience at DIS group was a real fresh air for my research which helped me to conclude my PhD. Looking forward to being back and starting the new adventure together!

I feel lucky to have met many wonderful people during my PhD who helped me in making this experience enjoyable and memorable. A big thanks to all the people at UCL, colleagues and staff, with who I just shared a coffee or a lunch in Cullem room. A special mention to all the people who became real friends. Starting with Hedaia who was there since the beginning, supporting and pushing me; thanks to you UCL started to be a second family. To Seph, with who I had the honour to start the LASP group, the time spent together at UCL excluding trips/internships etc. was very limited but so precious; thank you for all the good time, the gym time, the laughter, the endless coffees and Guinness and for our constructive conversations on diversity - you have always listened to me without prejudice helping me in understanding my limits and a way to improve. To Evi and Vasillis, for your

friendship and sweetness; to my friends in the office (Xinyue, Amany, Afroditi, Waseem, Andrea) and not (Dan, Vittorio, Callum and others ONG guys) for all the fun outside the office. To all the people from the gym, my second office!

I extend my thanks to all the members of DIS group in Amsterdam who were always extremely friendly and collaborative with me. An immense thanks to Irene, the friend this time! I don't even know how to explain it, our friendship was "at first sight" and you were my anchor in one of the most difficult moment of my life, *grazie di cuore*!

Then, there are all my home-based friends who have been always supportive and, waiting for me to be back and to spend time together as if I had never left. In particular, my deepest gratitude to my friend always present despite the distance, Sara, and her family, *la piccola e dolce* Ginny, Michele and Marty; *le ragazze di Gatteo*, Nadia, Karin, Eli with the little Lidia, Maru and Vale; to my childhood and beyond friend, Sara; to my first colleague-friend, Marco, with all our WhatsApp audios that have come with us from the master degree until today.

Last but not least, my deepest gratitude to all my family who supported and believed in me through this PhD. Including my chosen family Fede, the one who was my home in London: our adventure started by chance together and only we know the special bond it has gifted us - I couldn't even imagine something better! *Grazie amica mia*! And, my infinitive thanks to my parents whose unconditional love despite my choice brought me far from you has been my vital strength in this path and in life. Finally, to myself, to have followed my dream no matter what - you did it! Do not forget to be proud!

TENIAMO BOTTA
Silvia

Contents

List of Figures

List of Tables

List of Abbreviations

AR Augmented Reality

CAVE Cave Automatic Virtual Environment

CDN Content Delivery Network

CI Clique-Index

CMP Cube Map Projection

DASH Dynamic Adaptive Streaming over HTTP

DoF Degrees-of-Freedom

ERP Equirectangular Projection

FoA Focus of Attention

FoV Field of View

FP False Positive

FPR False Positive Rate

GBVS Graph-Based Visual Saliency

G-PCC Geometry-based PCC

HAS HTTP adaptive streaming

HEVC High-Efficiency Video Coding

HMD Head-Mounted Display

ILP Integer Linear Programming

IOC Inter-Observer Coungrency

IT Information-Theoretic

KNN K-Nearest-Neighbours

LiDAR Light Detection And Ranging

LR Linear Regression

LSTM Long Short-Term Memory

MI Mutual Information

MPD Media Presentation Description

MPEG Moving Picture Experts Group

MR Mixed Reality

MSE Mean Square Error

ODV OmniDirectional Video

OMAF Omnidirectional Media Application Format

PCC Point Cloud Compression

PSNR Peak Signal-to-Noise Ratio

QEC Quality Emphasis Centre

QoE Quality of Experience

RL Reinforcement Learning

ROC Receiver Operating Characteristic

RoI Region of Interest

SI Spatial Information

SRD Spatial Relationship Description

UAI User Affinity Index

UHD Ultra High Definition

V-PCC Video-based PCC

VQA Visual Quality Assessment

VoD Video on Demand

VR Virtual Reality

WLR Weighted LR

WMSE Weighted MSE

WS-PSNR Weighted Spherical PSNR

XR eXtended Reality

TE Transfer Entropy

TI Temporal Information

TP True Positive

TPR True Positive Rate

TSP Truncated Square Pyramid

Part I

FOREWORD

Chapter 1

Introduction

Over the past few years, the synergistic development of new mobile communication services (*i.e.*, fifth-generation (5G) mobile networks) and new cutting-edge portable devices (*i.e.*, smartphones) have helped for a breakthrough in video streaming services. The consumption of multimedia data on streaming platforms (*e.g.*, YouTube[1], Netflix[2]) or on social media (*e.g.*, Facebook[3], Instagram[4]) has become popular enough to play an important role in our day-to-day lives and to push developers to constantly make available new generations of video technologies [1, 2]. In this context, the concept of *immersive and interactive communication* is spreading, identifying a completely new way of communicating with others and displaying multimedia content. Traditional remote communications (*e.g.*, television, radio, video calling) are no more sufficient tools for our society: humans are inherently social, in need of realistic experiences, and traditional remote communications do not offer such full sense of immersion and a natural experience/interactions [3]. From here, the exploding research interest toward immersive technologies, with the ultimate goal of making remote communications as similar as possible to real face-to-face experiences. These technologies have landed in our everyday life with an impact of US$26.05 Billion in 2020 and with projection of growth to US$463.7 Billion by 2026 [4]. This will invest many sectors beyond entertainment, e.g., e-healthcare, e-education, and cultural heritage [5], since immersive communications address the compelling need of reducing the environmental impact and geographical barriers (or minority), enabling remote working and education and answering also natural emergencies needs (*e.g.*, reduced travel in pandemic, tornadoes, etc.).

The revolutionary novelty of immersive technology is to empower users with the possibility to interact with the digital elements (*i.e.*, objects or surrounding envi-

[1] https://www.youtube.com
[2] https://www.netflix.com
[3] https://www.facebook.com
[4] https://www.instagram.com

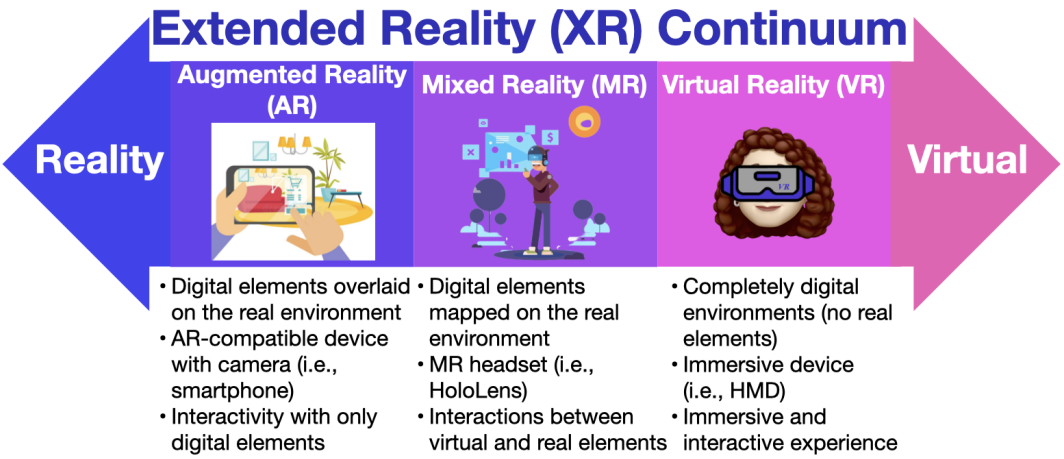


Figure 1.1: XR continuum of immersive technologies.

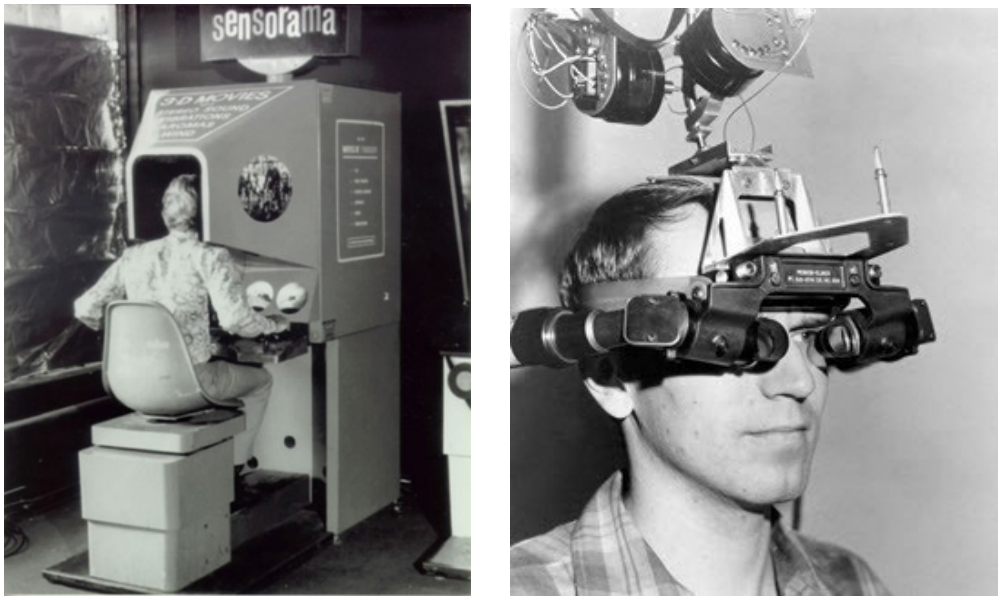
ronment), feeling engaged and present in a virtual space, even if they are not physically there. Specifically, eXtended Reality (XR) is the term that includes all current immersive technologies: VR, Augmented Reality (AR) and Mixed Reality (MR). As shown in Figure 1.1, these technologies fall in between full physical and virtual world realities. While AR and MR put together virtual and real objects respectively on a screen device and in the real world, VR let users immerse themselves in a virtual environment where they can navigate and interact.

The concept of *immersive and interactive communications* is spreading, identifying a completely novel way in which multimedia content is consumed. The user is clearly the main key factor for XR applications and consequentially the services need to be proper tailored to users. We are witnessing the beginning of a new *user-centric era*. This has opened to many new compelling open challenges, mainly due to the new role of the final user. Coding, storage, streaming and rendering need to be redesigned with the final goal of optimising the quality of the content displayed by the user, rather than the whole content quality and streaming service. With this goal in mind, a better understanding of how people interact with immersive content is needed and it is the focus of this thesis.

In the following, we briefly describe the main key features of the XR technology. Then, we show the main challenges that this new user-centric era has brought, highlighting the research questions that we aim to address in this thesis.

1.1 What is eXtended Reality (XR)?

Since the appearance of the first prototypes more than 60 years ago, XR technology has been increasingly developed. Even if XR technologies have different characteristics as shown in Figure 1.1, there are three revolutionary shared features: **presence**, **immersion** and **interact**. Presence refers to the illusory feeling experienced



(a) Sensorama Simulator. **(b)** First example of HMD.

Figure 1.2: First prototypes of XR devices.

by the user of being present in a virtual environment different from the physical one where they are actually located [6]. A condition necessary for presence is the immersion, which refers more to technical properties of the system that are needed to simulate realistic virtual environment [7]. Interactivity is instead the possibility for users to change the virtual environment with their movements [8]. Interaction is crucial to "feel present" in the virtual world: being able to move naturally helps the illusion of belonging to a different place. Novel types of multimedia content (*e.g.*, omnidirectional video and point cloud) are therefore needed to ensure a sufficient level of immersion, presence, and interactivity, which are the three crucial factors to guarantee high Quality of Experience (QoE) in a immersive system [9, 10]. Based on the selected immersive technology (*i.e.*, VR, AR, and MR), the levels of presence, immersion within a virtual space of which the user is empowered, and the enabled interactivity with this environment or virtual objects might be different [6, 11]. We now first briefly describe the history of XR, and then we provide more details and some examples of application per each immersive technology.

1.1.1 Brief History

Even if at the beginning there were no distinction between VR,AR, and MR, the first concept related to what we now call now XR appeared in 1935 in a science fiction book, "Pygmalion's Spectacles" by Stanley Weinbaum [12, Chapter 2]. The main character wearing special eyeglasses equipped with sensors experiences an alternative world replacing real-world stimuli with artificial ones. For the first time,

(a) Facebook Horizon [16].

(b) Immersive VR museum [17].

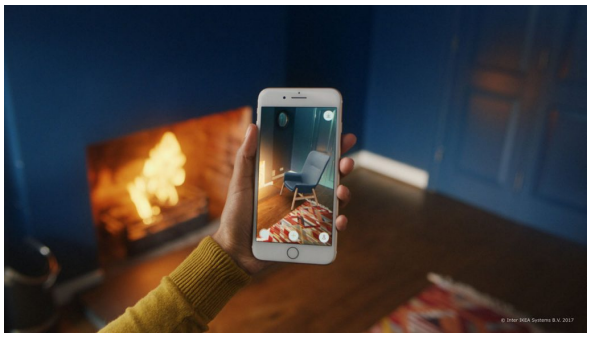
Figure 1.3: Examples of VR applications.

the protagonist has the illusion to *be immersed* and *present* in a different environment from the real one. Gradually, first prototypes of XR started to appear: from the first world-fixed displayed for an immersive movie, named *Sensorama* (Figure 1.2 (a)) and designed by Morton Heiling in 1956 [13] to the first Head-Mounted Display (HMD) with head tracking and able to render computer-generated images in 1965 by Ivan Sutherland [14] (Figure 1.2 (b)). Only in 1987, the term *virtual reality* has been coined by John Lanier and VR gadgets (*i.e.*, commercial HMD, tactile gloves with optical sensors) appeared on the market. With ages, technologies have improved and brought to a real explosion at the beginning of the 21st century such that now it is envisioned that XR could radically transform our lives and work in the next decade [15]. The term XR embraces all the immersive technologies developed until now aimed either at creating a fully immersive experience such as VR or at combining the virtual with the reality, as for AR and MR. To avoid misunderstanding, we now briefly describe their main characteristics bringing also some real example of application per each technology.

1.1.2 Virtual Reality (VR)

Virtual Reality (VR) is the first example of XR been developed. This technology refers to a fully digital environment that replaces the real world and in which the user is immersed. This digital environment allows the user to experience a completely new reality. While traditional videos are passively consumed by viewers, VR content let any final user actively navigate in the scene, providing a sense of full immersion within a virtual environment. The interest in this application is going so fast that more and more companies nowadays are investing it. The gaming sector has been the first to have shown an exploding interest adding many new VR gadgets to their consoles such as Nintendo Labo [18] and Sony Playstation VR [19]. But the potential of VR does not stop here. For instance, Facebook has recently launched *Horizon* a new social platform where people can immerse themselves and meet with

(a) Pokémon Go app [22].



(b) Ikea app [24].

Figure 1.4: Examples of AR applications.

other friends in a virtual space (Figure 1.3 (a)). The Museum of Monte San Michele, near Gorizia (Italy), takes advantage of VR technologies to allow people to live virtually historical events of the First World War (Figure 1.3 (b)). Also governments are showing their interest on this technology by supporting digital innovation in museums [20].

1.1.3 Augmented Reality (AR)

In contrast with VR where the real world is completely replaced by a virtual environment, Augmented Reality (AR) offers a new perception of *real* elements with a combination of *virtual* objects. This technology shows on the same screen device (*i.e.*, smartphone) computer-generated elements overlapped to the reality that surrounds the viewers and enables real-time interaction with the digital objects [21]. Therefore, AR enriches the user experience in the real world adding extra information. The main advantage of this technology is that can be experienced through common devices and indeed there are already some popular AR applications in our daily life. The Pokémon GO app [22] is the first mobile AR game that exploded in popularity in 2016 becoming a real trend [23]. As shown in Figure 1.4 (a), it is a location-based game that renders cartoon characters (*i.e.*, Pokémon) onto the real world via smartphone technology. Also Ikea has developed in its app an AR functionality that allows the consumer to display virtual furniture placed in your physical room (Figure 1.4 (b)).

1.1.4 Mixed Reality (MR)

The last example of XR technology is a combination of the previous two. Mixed Reality (MR) indeed joins both physical and digital elements allowing interactions among them [25]. Similarly to VR applications, a specific device is needed to experience MR: holographic eyeglasses device. In contrast to HMD, these eyeglasses do not block the vision of the surrounding environment but place volumetric digital

(a) AR Dynamics 365 Remote Assist [27].

(b) Envisioned MR scenario in everyday life [28].

Figure 1.5: Examples of MR applications.

content as if it were in the real environment. Similarly to AR, the viewer is enabled to interact with the virtual objects but in this case through physical movements in the real space. HoloLens is an example of MR headset developed by Microsoft [26]. Even if there is a growing attention for this technology, it is still at its infant stage. Figure 1.5 shows two examples of MR use cases: a remote assistant that helps technicians in working together from different locations (Figure 1.5 (a)) and a futuristic scenario with several elements of MR aimed at helping people in their daily routine (Figure 1.5 (b)).

1.2 Main challenges and research questions

In the context of this thesis, we mainly focus on Virtual Reality immersive streaming applications. The interest for VR technology is envisioned to explode in the near future: a huge annual growth (around 18.0% compound annual growth rate) of sales is indeed forecast from 2021 to 2028 impacting on quality of life, environmental and energy conservation, and world economy [29]. Immersive reality technology has revolutionised how users engage and interact with media content, going beyond the passive paradigm of traditional video technology, and offering higher degrees of presence and interaction in a virtual environment. In fact, the key novelties of VR systems is to guarantee immersion, presence and interactivity to any final users in order to make the immersive experience real as much as possible [30]. Moreover, depending on the enabled locomotion functionalities in the 3D space, VR environments can be classified as 3- or 6-DoF. In the first scenario, the de-facto multimedia content is the *OmniDirectional Video* (OmniDirectional Video (ODV)) (also named 360° or spherical video) which represents an entire 360° environment on a virtual sphere. The viewer is fully immersed in a virtual space where they can navigate and interact thanks to an immersive device – typically an HMD, which enables to display only a limited portion (*i.e.*, Field of View (FoV)) of the environment around him/herself, named *viewport*. The media is displayed from an *inward*

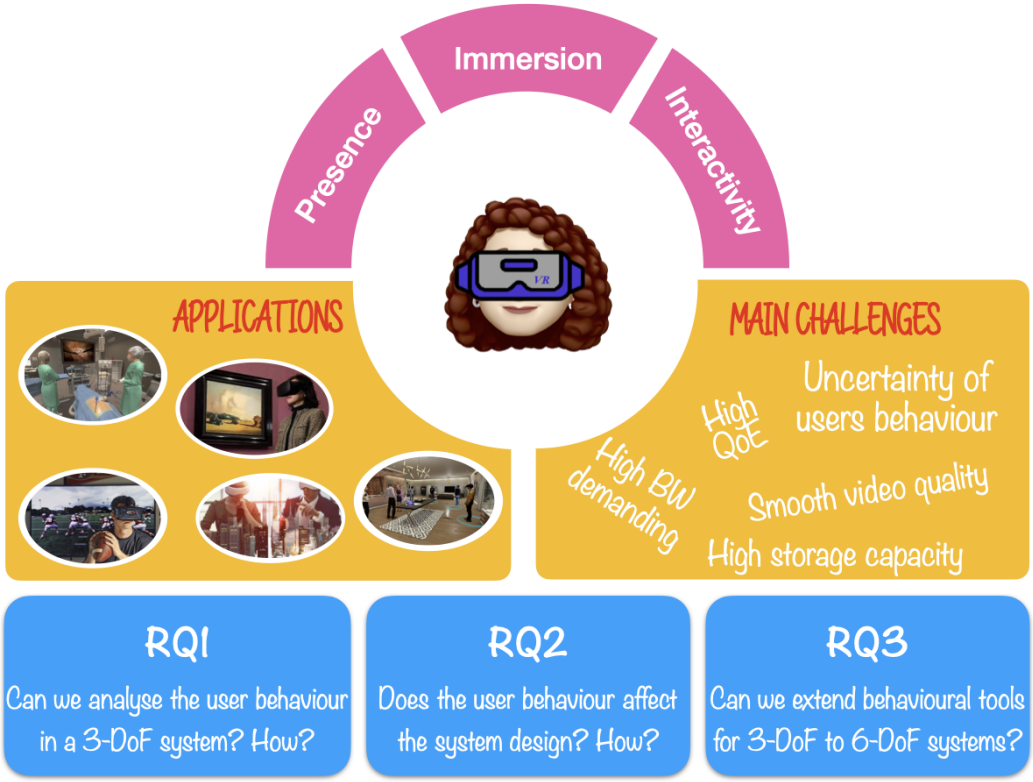


Figure 1.6: Main challenges in VR systems and research questions addressed in this thesis.

position, and the viewer can interact with the content only by changing the viewing direction (*i.e.*, by looking up/down or left/right or tilting the head side to side). In a 6-DoF system, the user can also change viewing perspective by moving (e.g., walking, jumping) inside the virtual space. The scene is therefore populated by *volumetric objects* (*i.e.*, meshes or point clouds) which are observed from an *outward* position.

Despite their differences, the common denominator of both these immersive and interactive systems is the viewer as an active decision-maker of the displayed content. This active role of the user defines the *user-centric* era (Figure 1.6), in which content preparation, streaming, and rendering need to be tailored to the viewer interaction to remain bandwidth-tolerant whilst meeting quality and latency criteria. These requirements imply a very high amount of data to be transmitted in real time for the millions of VR users envisioned in the near future, pushing also connectivity boundaries. Thus, the novel type of immersive content and the interactive way of consuming media data have raised several new challenges and implications in the context of interactive video streaming which can be summarised as follow:

- **uncertainty of users behaviour**;

- **highly data intensive system** (large volume of data to be stored and delivered over bandwidth-limited network);
- **smooth video quality to ensure high QoE**;
- new immersive multimedia content to capture and process;
- viewport extraction in real time based on the users interaction (*i.e.*, low latency);
- ultra-low complexity requirements;
- interactive and immersive rendering at the client side.

For the sake of completeness, the main challenges related to VR immersive streaming have been mentioned. However, in this work, we tackle the first three aforementioned challenges (highlighted in bold in the above list). Our main goal is to understand how people interact with the immersive content and to study the impact of the user behaviour on the system design in novel user-centric solutions. These studies have been first conducted considering only navigation in 3-DoF environments to finally be extended to a more challenging systems, such as 6-DoF. In summary, in this thesis, we address the following main research questions:

RQ 1 *Can we analyse the user behaviour in a 3-DoF system? How?*

RQ 2 *Does the user behaviour affect the system design? How?*

RQ 3 *Can we extend behavioural tools for 3-DoF to 6-DoF system?*

1.3 Main contributions

We now summarise the main contributions developed when addressing three research questions presented in the previous section.

Behavioural analysis in 3-DoF VR

The first contributions of this thesis are aimed at better understanding and enabling behavioural analysis of users navigating in a 3-DoF VR environment. Specifically, the main contributions aimed at addressing **RQ1** are the following:

1. An exhaustive **state-of-the-art on behavioural analysis in VR application** and the role of the user in **coding and streaming solutions**. The overview explores how the user navigation has been analysed in the literature and which

tools have been developed to predict the user behaviour. We also present the main solutions to improve and optimise immersive coding and streaming systems, with a novel focus on *user-centric streaming solutions*. This work will be published in [31] and is presented in Chapter 3 of this document.

2. A novel **graph-based method** to identify **clusters** of users who are attending the same portion of the spherical content over time. The proposed solution takes into account the spherical geometry of the content and aims at clustering users based on the actual overlap of displayed content among users. This work was published in [32] and appears in Chapter 4 of the thesis.

3. A **publicly available dataset** complementing existing current ones that provides navigation trajectories acquired for heterogeneous omnidirectional videos and **different viewing platforms**, namely, head-mounted display, tablet and laptop. We also present an exhaustive analysis on the collected data, to better understand navigation in VR across users, content, and for the first time across viewing platforms. A novelty lies in the **user-affinity metric** proposed to investigate users' similarities when navigating within the immersive content. The analysis reveals useful insights on the effect of both device and content on the navigation, which could be precious considerations from the system design perspective. This work was published in [33] and appears as a case of study in Chapter 4 in this thesis.

4. To adopt **trajectory-based metrics** from information theory to VR domain highlighting the importance of looking at users trajectories instead of more qualitative measures of user's interactions. In particular, we propose two line of investigations: **intra-user behaviour analysis**, aimed at understanding the level of interactivity of each single user across different content, and **inter-user behaviour analysis** which considers navigation across an entire group of viewers to asses if user's behaviour can help in the prediction of other viewer's behaviour. This work was published in [34] and is described in Chapter 5.

User-centric 3-DoF VR system

The second part of contributions in this thesis is aimed at answering the research question **RQ2** and thus, understanding how taking into account the user behaviour can help in enabling novel user-centric solutions. To this end, the main contributions of this second part are the following:

1. A case study on **user-centric optimisation for coding and storing** 360° video at the server. The main aim is to find the optimal coding parameters

that optimise the QoE perceived by the final users minimising the storage and coding costs. The key novelty is to consider the users' behaviour and network characteristics. We formulate an integer linear program that seeks the best stored set of omnidirectional content that minimises encoding and storage cost while maximises the user's experience. This is posed while taking into account network dynamics, type of video content, but also user population interactivity. This work led to publication [34] and is presented in Chapter 6 of this document.

2. A case study on **navigation-aware delivery strategy** for 360° video system. The key novelty is to take into account the users behaviour while downloading the video content over limited networks. This work was published in [35] and appears in Chapter 7 of this thesis.

Towards behavioural analysis in 6-DoF VR

The last part of this thesis moves the focus of attention to 6-DoF VR systems. Aimed at answering the research question **RQ3**, the main contributions presented in this thesis are the following:

1. To extend the **applicability of tool for investigating behavioural similarity** (*i.e.*, users sharing common behaviour while interacting with virtual content) **from 3-DoF to 6-DoF environment**. To do so, we first investigate how new physical settings and locomotion functionalities given to users can affect the analysis and understanding of their behaviour, highlighting the main limitations of existing tool when applied to 6-DoF. Then, we propose a new methodology for overcoming those limitations defining **novel metrics** aimed at **capturing users trajectory similarity**. A first intuition of this work was published in [36] to be then extended in [37] while in this thesis is presented in Chapter 8.

2. A case of study on **behavioural analysis of user navigating in 6 degrees of freedom social VR movie**. We mainly investigate how users are affected by salient agents (*i.e.*, virtual characters) and by narrative elements of the VR movie (*i.e.*, dialogues versus interactive part). This work was published in [38] and appears in Chapter 9.

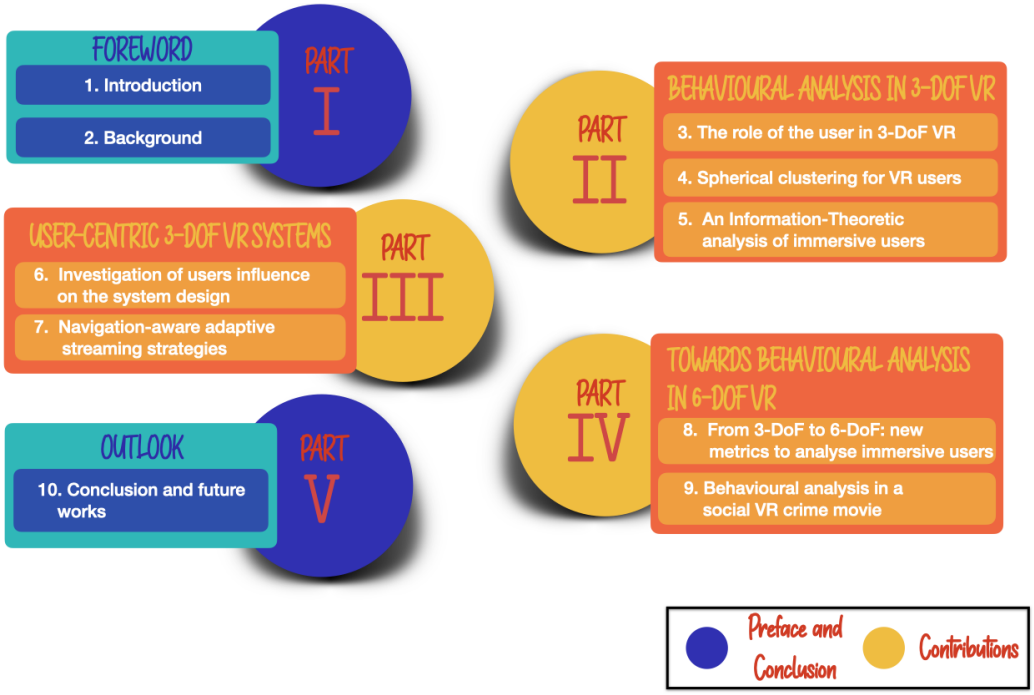


Figure 1.7: Structure of this thesis.

1.4 Outline

This thesis consists of ten chapters divided in five parts. As shown in Figure 1.7, two parts are preface and conclusions, while three are novel contributions. Subsequent to this introductory chapter, the rest of this document is organised as follows:

- **Chapter 2** introduces the relevant background knowledge related with immersive communication system to establish the importance and to better understand the work done as part of this PhD thesis. The chapter starts with definitions of user navigation trajectory in both 3- and 6-DoF environments. A detailed overview of the immersive streaming pipeline is also presented.

- **Chapter 3** presents a comprehensive survey on the role of the user in 3-DoF VR system identifying the place where most of the new contributions of this PhD thesis are made and significant. The chapter starts with an overview of existing surveys related with ODV streaming systems and publicly available navigation dataset. Then, an in-depth overview of the research efforts done to analyse user behaviour while navigating in a VR content is presented. To conclude, the chapter shows how behavioural information has been leveraged to advance ODV streaming strategies in the latest user-centric systems.

- **Chapter 4** describes a novel tool to identify groups of users who behave in a similar way while interacting with virtual content. Specifically, a graph-based

clustering algorithm is proposed, which takes into account the spherical geometry of the content and aims at clustering users based on the actual overlap of displayed content among users. We also introduce a new metric, namely the *User Affinity Index*, which quantifies users' similarity based on the proposed clustering tool. We then present an extensive behavioural data analysis across content *and* across viewing device, carried out on a new dataset collected in collaboration with Trinity College Dublin.

- **Chapter 5** introduces a novel behaviour analysis in a 3-DoF VR scenario aimed at characterising navigation patterns not only across users but also across content. Namely, this chapter presents an *intra-user behavioural analysis* focused on understanding the behaviour of each individual when navigating in VR, and an *inter-user behavioural analysis* aimed at understanding how much information about a single content can be extracted when observing an entire population of viewers. This is carried out by considering a space-time trajectory domain rather than only a spatial domain, and translating the concept of entropy into users trajectory predictability.

- **Chapter 6** presents a first attempt of investigation of behavioural influence on the system design. Specifically, a novel *user-centric immersive algorithm* is defined to optimise the set of ODV representations to be stored at the server, minimising the total cost and yet maximising the final quality. The key-novelty of this algorithm is to take into consideration users' behaviour beyond the spherical geometry and content information.

- **Chapter 7** proposes an optimal transmission strategy for 3-DoF VR applications able to fulfil the bandwidth requirements, while optimising the end-user quality experienced in the navigation. Specifically, we consider a tile-based codec content for adaptive streaming, and a novel *navigation-aware* transmission strategy at the client side (*i.e.*, adaptation logic) is presented. The novelty lies in considering both a viewport-quality as metric that reflects the quality of any portion of the sphere displayed by the end-user but also the popularity of each viewport to be displayed through *heatmaps*.

- **Chapter 8** investigates navigation trajectories of users within a 6-DoF VR environment. The work in this chapter is aimed at enabling user behavioural analysis in the case of 6-DoF systems by extending the applicability of existing behavioural methodologies adopted for the 3-DoF counterpart. After motivating the need for this study, we developed novel similarity metrics taking into account the new physical settings and locomotion functionalities given

to users in 6-DoF . The chapter ends with a case of study which points out the robustness and versatility of these metrics since they preserve good performance on navigation trajectories collected in a 6-DoF AR scenario.

- **Chapter 9** provides a case of study of behavioural analysis of users navigating in a 6-DoF social VR movie. Specifically, navigation trajectories from a photorealistic telepresence experiment, in which subjects watch a crime movie together in VR, are analysed. The novelty is to investigate how users are affected by virtual characters and by narrative elements of the movie (*i.e.*, dialogues versus interactive part).

- **Chapter 10** summarises the work presented in this thesis and highlights the importance of the proposed methods and tools. In addition, the chapter outlines future research directions.

1.5 Publications

The research presented in this document has resulted in 2 journal publications, 1 invited book chapter, 5 international conference publications, 1 invited talk, and 3 awards, listed below in chronological order:

- *Journal Publications*

    1. **Silvia Rossi**, Cagri Ozcinar, Aljosa Smolic, and Laura Toni "Do Users Behave Similarly in VR? Investigation of the User Influence on the System Design". Transactions on Multimedia Computing, Communications, and Applications (TOMM), ACM, 2020. doi:/10.1145/3381846.

    2. **Silvia Rossi**, Irene Viola, Laura Toni, and Pablo Cesar "From 3-DoF to 6-DoF: New Metrics to Analyse Users Behaviour in Immersive Applications." *Submitted to* Transactions on Image Processing, IEEE, 2021. doi:https://arxiv.org/abs/2112.09402

- *Book Chapter*

    1. **Silvia Rossi**, Alan Guedes, and Laura Toni "Coding, Streaming, and User Behaviour in Omnidirectional Video." *To be published in* Immersive video technologies. Elsevier, February 2022.

- *Conference Publications*

    1. **Silvia Rossi** and Laura Toni "Navigation-aware adaptive streaming strategies for omnidirectional video". Proceedings of 19th International

Workshop on Multimedia Signal Processing (MMSP), IEEE, 2017. doi:10.1109/MMSP.2017.8122230.

2. **Silvia Rossi**, Francesca De Simone, Pascal Frossard and Laura Toni " Spherical clustering of users navigating in 360-degree content.", Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), IEEE, 2019. doi:10.1109/ICASSP.2019.8683854.
3. **Silvia Rossi** and Laura Toni "Understanding user navigation in immersive experience: an information-theoretic analysis". In Proceedings of the 12th International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE), ACM, 2020. doi:10.1145/3386293.3397115.
4. **Silvia Rossi**, Irene Viola, Jack Jansen, Shishir Subramanyam, Laura Toni, and Pablo Cesar "Influence of Narrative Elements on User Behaviour in Photorealistic Social VR". In Proceedings of the 13th International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE), ACM, 2021. doi:10.1145/3458307.3463371.
5. **Silvia Rossi**, Irene Viola, Laura Toni, and Pablo Cesar "A new Challenge: Behavioural analysis of 6-DoF user when consuming immersive media". In Proceedings of International Conference on Image Processing (ICIP), IEEE, 2021. doi:10.1109/ICIP42928.2021.9506525.

- *Others Conference Publication*

1. Pedro Gomes, **Silvia Rossi** and Laura Toni, "Spatio-Temporal Graph-RNN for Point Cloud Prediction". In Proceedings of IEEE International Conference on Image Processing (ICIP), IEEE, 2021. doi:10.1109/ICIP42928.2021.9506084.

- *Invited talk*

1. **Silvia Rossi** and Laura Toni, "Adaptive Streaming for Immersive Communication." Technical presentation at 57th FITCE Congress - Delivery and Consumption of Digital Media, Salford (UK), 2018. (https://www.youtube.com/watch?v=Um1BsjDciZI&t=19s).

- *Awards & Scholarships*

1. **2018 Cisco Institute Prize for best student poster**, Mildner Memorial Lecture, UCL. Poster title: "*Adaptive Streaming for Immersive Communication*".

2. **2021 Nicholas Georganas ACM TOMM Best paper award** with "*Do Users Behave Similarly in VR? Investigation of the User Influence on the System Design*". ACM TOMM Volume 16, Issue 2, June 2020, Article N. 46, pp 1-26.
3. **2021 ERCIM Alain Bensoussan Fellowship Programme.** An outstanding scholarship for funding one year as postdoctoral research at CWI, The Netherlands.

Chapter 2

Background

This chapter introduces the relevant background knowledge related to immersive communication systems. To ensure a good understanding of the body of this dissertation, we first review the role of the user while navigating within immersive content defining their navigation trajectory and the differences between a 3- and 6-DoF environment (Section 2.1). Then, we briefly overview the main characteristics of the immersive streaming pipeline in Section 2.2. Finally, we describe the most popular solutions adopted for video adaptive streaming in both a 3-DoF and 6-DoF environment in Section 2.3 and 2.4, respectively.

2.1 User Navigation in immersive systems

Virtual reality and immersive technologies at large have been revolutionising how users engage and interact with the media content, going beyond the passive paradigm of traditional video technology, and offering higher degrees of immersiveness and interaction. VR technology refers to a fully digital environment that replaces the real world and in which the user is immersed. Depending on the enabled locomotion functionalities in the 3D space, VR environments can be classified as 3- or 6-DoF. We now describe these VR environments (both 3- and 6-DoF) highlighting the main features of their immersive content and defining the navigation trajectories of user.

2.1.1 Navigation in a 3-DoF VR Environment

The de-facto multimedia content for a 3-DoF experience is an *ODV* (also named 360° or spherical video), which acquires a 360° scene instantaneously and promises an immersive and interactive experience. In this scenario, the viewer is placed at the centre of the virtual space (*i.e.*, viewing sphere) and provided with a VR device – typically a Head-Mounted Display (HMD) – experiences a 3-Degree of Freedom (DoF) interaction with the content, by looking up/down (pitch) or left/right (view) or by tilting their head from side to side (roll), as shown on the left box of Figure 2.1.

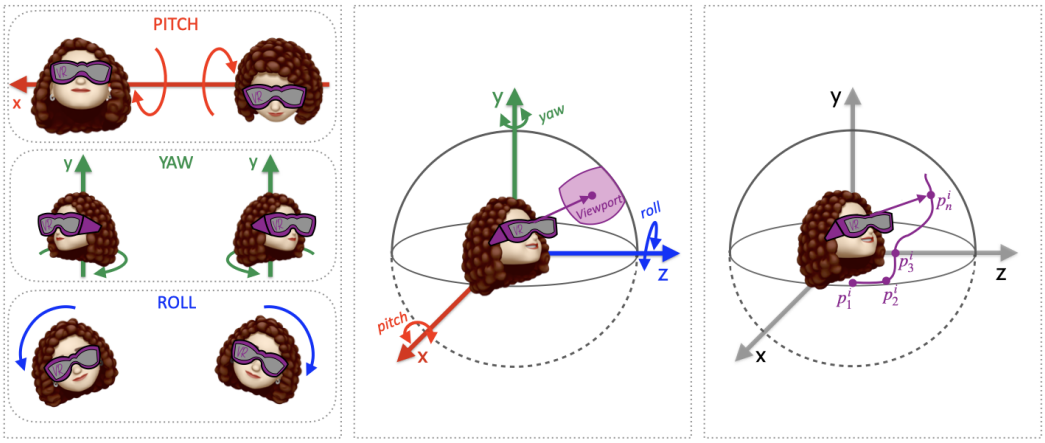
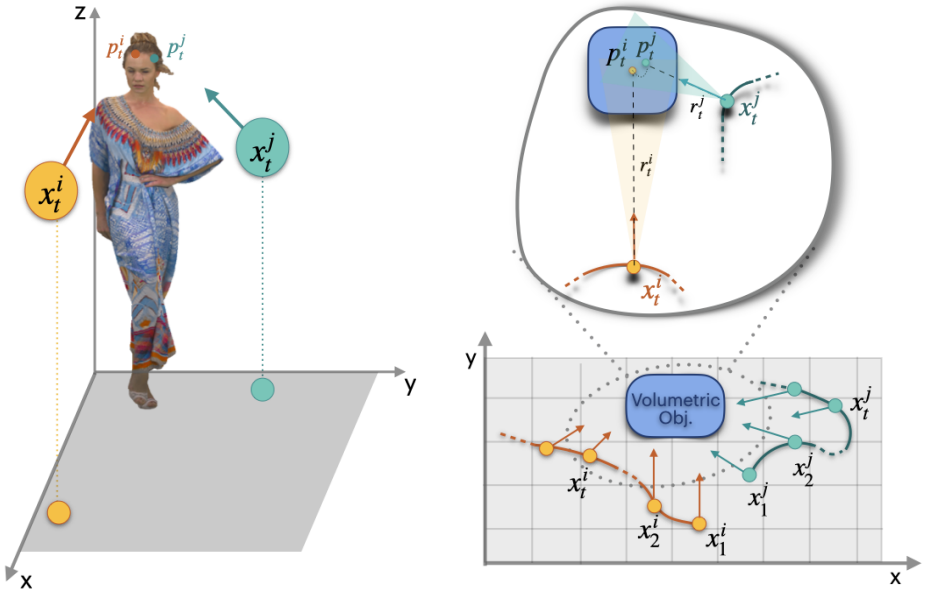


Figure 2.1: Navigation in immersive content. From the left to the right box: rotational head movements (pitch, yaw and roll); navigation system on viewing sphere at generic time instant; navigation trajectory on the sphere over time.

These are rotational movements around x-, y- and z-axes, named respectively *pitch*, *yaw*, and *roll*. To mimic a real-life scenario, the user cannot display the entire environment around him/herself, but only a restricted FoV of the environment around themselves, named *viewport*, identified by the viewing direction at any given time (central box of Figure 2.1). Hence, the sequence over time of the user viewing direction can be used to identify the user behaviour in an immersive experience. In particular, the viewing direction can be approximated by the projection of the viewport centre on the viewing sphere. More formally, we can define the *navigation trajectory* of a generic 3-DoF user i as $\{p_1^i, p_2^i, .., p_n^i\}$ where p_t^i is the center of the viewport projected on the immersive content at a given timestamp t (right box in Figure 2.1). The point p can be represented in spherical coordinates by $[\theta_p, \phi_p, r]$ where $\theta_p \in [-\pi, \pi]$ is the azimuth angle (or longitude), $\phi_p \in [-\pi/2, \pi/2]$ the polar angle (or latitude), and r is the distance between the point (viewport center projected on the immersive content) and the origin (user position). The constant distance r between the user and the media content is a key feature that makes the position of the viewport centre alone highly informative about user interaction and similarity among users.

2.1.2 Navigation in a 6-DoF VR Environment

As just seen, a 3-DoF experience provides the user with some basic level of interaction within the content (*i.e.*, selecting the displayed viewport based *only* on the viewing direction). To augment the sense of immersion, a higher level of interactivity is provided in a 6-DoF. Here, the enabled actions given to viewers are extended to translation movements: the user is now free to naturally walk and jump inside the virtual space beyond the previous rotational head movements showed in Figure 2.1.



(a) Navigation of user i and j represented in 3D space.

(b) Navigation trajectories projected in a 2D domain for user i and j.

Figure 2.2: Navigation in a 6-DoF system.

The higher level of interaction makes the 6-DoF experience more immersive.
A new immersive media content is also needed: since the scene can be viewed from any direction and angle at any moment in time, there is the need to add an extra dimension to the virtual environments, the *depth*. Therefore, the typical media for a 6-DoF experience is a *volumetric content* (*i.e.*, point clouds or meshes) which represents 3D objects or scenes as a collection of spatial points. Figure 2.2 (a) gives an example on how this content is visualised in the 3D space by two different users. It is already clear that the more degrees of freedom are given to the user, the more challenging becomes the system and also the description of user navigation within it. The viewport centre alone is no more sufficient to characterise the user behaviour since not only the viewing direction but also the distance between the user and the immersive content can change over time. To better understand this concept, in Figure 2.2 (b) we depict an example of two users navigating in a 6-DoF system projected into a 2D domain (*i.e.*, floor). In the bottom part of Figure 2.2 (b), the navigation trajectories of two users i and j are shown. Each point x_t represents the spatial coordinates (*i.e.*, [x,y,z]) on the floor while each associated vector symbolises the viewing direction. In the top part of Figure 2.2 (b), we have instead a zoom-in snapshot at a specific time instant t. In details, the user's *viewing frustum* is represented by triangles, which indicate the area within the user's viewport and r_t^i is the distance between user and the volumetric content. We have also depicted the viewport center p_t^i projected on the displayed volumetric object. Given the users i

Table 2.1: Immersive streaming historical timeline.

2007	Google launches Street View [39]
2011	ISO publishes MPEG-DASH [40]
2014	Google launches Cardboard and Facebook acquires Oculus [41, 42]
	VRChat is released for the first time as Window application [43]
2015	YouTube and Facebook social platforms allow ODV upload [44, 45]
	MPEG standardised MPEG-DASH SRD to support tiled streaming [46]
2016	BBC and ARTE begin share ODV content [47, 48]
	The first prototype of Hololens is introduced by Microsoft [49]
2017	Vimeo platform allows ODV upload [50]
2018	MPEG starts working on MPEG-I for immersive media [51]
2019	OpenXR, a platform for XR software development, is released [52]
2020	Apple launches iPad Pro and iPhone 12 Pro with build-in LiDAR sensors [53]
2021	Mark Zuckerberg announces Facebook vision towards the metaverse [54]
	Microsoft introduces Mesh for Microsoft Teams [55]

and j at time t with $r_t^i \gg r_t^j$, the user j, who is very close to the object, will visualise a very focused and detailed part of it; conversely, user i is pointing to the same area but from further distance, thus she/he will experience the content differently. It is worth noting that, even if the displayed content is quite diverse, the viewport centres p_t^i and p_t^j are very close. This highlights that the viewport centers alone is not sufficient anymore to represent the user within a 6-DoF environment. The distance r between the viewer and the object is now crucial to identify the actual displayed portion of the content. Therefore, we define the navigation trajectory for a generic 6-DoF user i as $\{(x_1^i, p_1^i, r_1^i), (x_2^i, p_2^i, r_2^i), \ldots, (x_n^i, p_n^i, r_n^i)\}$. In addition to the viewport center p_t^i projected on the displayed volumetric object, there are also x_t^i which represents the spatial coordinates (*i.e.*, $[x,y,z]$) of the user in the VR environment and the distance r_t^i as the distance between user and viewport center p_t^i.

In this section, we have formally defined how to track of user interactivity in both a 3- and 6-DoF environments by navigation trajectories. These will be used in the remaining body of this thesis.

2.2 Immersive Communication Pipeline

In this section, we provide an overview of a generic immersive streaming pipeline shown. We start with an historical overview to contextualise the first steps that opened the gate to immersive video streaming research, we then overview the key components of the streaming pipeline, from acquisition to rendering, for both 3- and 6-DoF setting.

Table 2.1 depicts the historical evolution that led to current technology used for immersive systems. This evolution has been characterised by three key compo-

nents: 1) large-scale utilisation of immersive applications (blue events in Table 2.1); 2) immersive displaying technology (green events); 3) technological advances in the streaming pipeline (purple events). One first service that appeared in 2007 based on omnidirectional content was the Google Maps Street View, which allows users to virtually navigate on streets using a sequence of omnidirectional images [39]. After this, the ODV market has grown significantly mainly when YouTube and Facebook (and Vimeo) allowed the upload and share of 360-degree content on their platforms in 2015 (in 2017) [44, 45, 50]. On parallel, a first example of social VR platform, VRchat, is released in 2014 moving a step forward in the immersive experiences. The interest in immersive systems then has been grown exponentially: for example, BBC and the French cultural network ARTE used 360-video for immersive documentaries. Moreover, a new cross-platform aimed to facilitate the development of immersive content named OpenXR has been released in 2019. Nowadays, 360-degree content is widely used across multiple sectors (*e.g.*, e-culture, entertainment, retail, live sports) amplified even further from recent attention to metaverse applications by technological giants such as Facebook and Microsoft [54, 55].

This widespread of immersive services was further pushed by the advances on screen devices: in 2014 Google proposed a very cheap mobile-based HMD called Cardboard, while Facebook made a two-billion-dollar acquisition of the HMD company Oculus. Also, the vision and creation of volumetric content have been made more accessible respectively in 2016 when Microsoft released the first Hololens (mixed reality smart glasses), and in 2020 when Apple put on the market the first mobile devices with Light Detection And Ranging (LiDAR) sensors (device to capture 3D environment).

These improvements in the technologies have led to a ever-growing desire for the users to experience immersive content, highlighting the compelling need for research advances and even standardising steps on immersive streaming pipeline. The well known Moving Picture Experts Group (MPEG)-Dynamic Adaptive Streaming over HTTP (DASH) –de-facto streaming solutions standardised in 2011 [40]– has been improved to enable immersive systems. At first, DASH streaming was extended to the tile-based encoding that has played a key role in viewport dependent streaming of omnidirectional content. In more detail, the ODV is spatially cropped in different bitstreams named tiles, each of those independently coded from the other tiles, allowing for unequal quality levels [56]. Tiles from different encoding quality can therefore be combined in a single bistream such that only a single decoder is required for the playback. The other key aspect of tile-based streaming is the DASH *Spatial Relationship Description* (SRD) [46], which enables the transmission of only a portion of the video. This, in combination to multi-quality tile

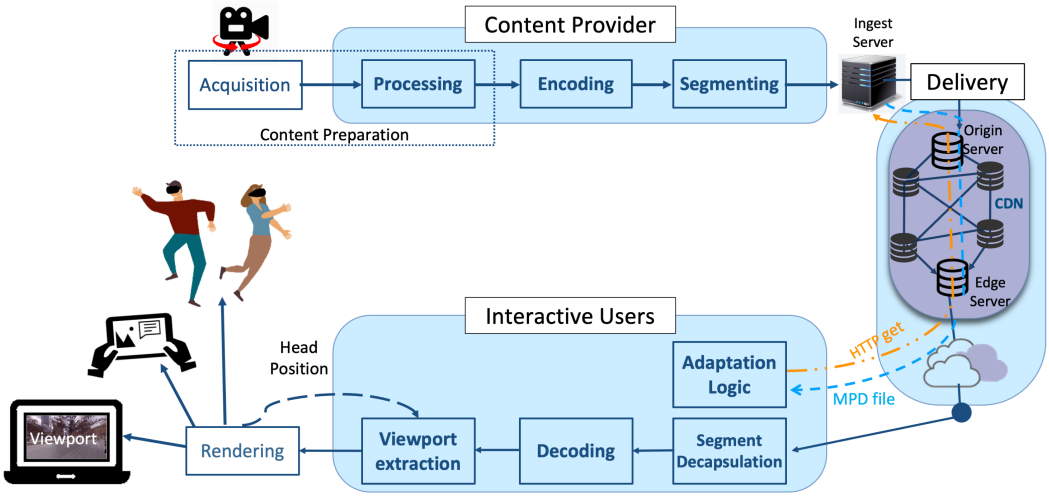


Figure 2.3: Immersive video streaming pipeline.

based coding allows us to send at high quality only the portion of interest to the VR user. This will be a key advance in viewport-dependent streaming technologies for ODV (discussed in Section 2.3.2). These above interests for 360-degree content were then consolidated in 2018 when MPEG started working on a new era of standardisation for immersive applications with MPEG-I (I stands for immersive). The new effort of MPEG is to have the first international standard for storage and distribution of immersive content including ODV and volumetric content to free navigate in a 3D space.

These technological and standardisation advances pushed research efforts to improve even further the immersive pipeline to achieve better services in terms of bandwidth, storage, networking caching, and perceived user quality. In the following, we describe the main components of immersive delivery pipeline (focusing mainly on MPEG-DASH protocol[1]) from acquisition to rendering highlighting how this has been adapted from classical 2D video to immersive streaming. Given the similarities, we consider a general pipeline shown in Figure 2.3 for both 3- and 6-DoF VR system (for spherical and volumetric content, respectively). Then, in the following sections we provide an overview on the main technological advances mainly from the coding and streaming perspective. Initial efforts were mainly focused on ODV system-centric streaming, see Section 2.3. Recently, some researchers focused their studies on enabling also the emerging 6-DoF VR system, see Section 2.3.

[1]It is worth mentioning that other streaming protocols (not purely DASH based) have been proposed for ODV [57], however we mainly focus on DASH advances as this conceptually covers the majority of the works.

(a) Camera ring [58]. **(b)** Dual Fisheye [59].

Figure 2.4: Examples of omnidirectional camera.

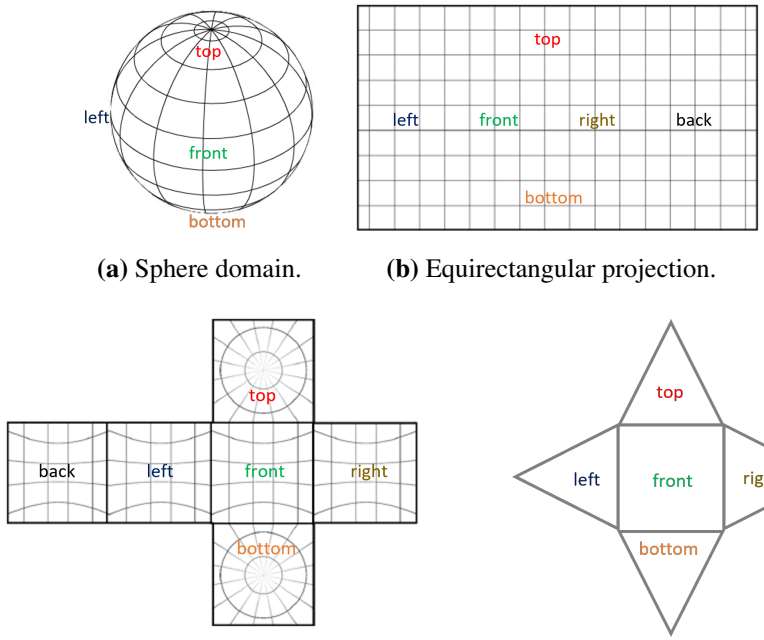
2.2.1 Content Preparation and Compression

The initial step to enable an immersive VR experience is *Content Preparation*, composed of acquisition and processing, which refers to potential media manipulation before the compression. As for every multimedia content before being transmitted, the next step is the *Compression* (encoding/decoding). As defined at the beginning of this Chapter in Section 2.1, immersive VR systems can be classified as 3- and 6-DoF environment and they are characterised by specific multimedia content, ODV and volumetric video. In the following, we briefly describe the content preparation and compression phases for both these immersive formats, highlighting also the main challenges.

Omnidirectional Content

The most popular and practical devices to capture omnidirectional videos are camera ring and omnidirectional camera. Figure 2.4 (a) shows an example of camera ring, composed by multiple cameras able to acquire the scene simultaneously from different directions covering 360° angle. Figure 2.4 (b), instead, depicts a simpler omnidirectional camera with two eye-fish cameras recording ultra wide-angle videos. In both cases, the output is a single omnidirectional video stream obtained by combining (*i.e.*, stitching [60]) and synchronising the content acquired by the different cameras. Stitching and synchronisation of the spherical video are very challenging processes and they can drastically compromise the quality of the content. Moreover, the spherical video needs to be recorded at high resolution and frame rate since most probably it will be displayed by the HMD and therefore very close to the human eye.

To be manipulated by existing 2D media processing tools, the spherical signal (Figure 2.5 (a)) has to be projected on the planar domain obtaining a *panoramic* version. This projection phase is part of *Processing* step for ODV in Figure 2.3. The most commonly employed sphere-to-plane projections are the Equirectangular Projection (ERP) (Figure 2.5 (b)), Cube Map Projection (CMP) (Figure 2.5 (c)) [61]



(a) Sphere domain. **(b)** Equirectangular projection.

(c) Cube Map projection. **(d)** Pyramid Projection.

Figure 2.5: Most popular map projections from sphere to planar domain.

and Pyramid Projection (Figure 2.5 (d)) [62]. However, it is well known that these projects introduce artefacts affecting the final video quality. We now briefly describe these projections to highlight the main differences:

Equirectangular Projection The simplest and most popular sphere-to-plane mapping is the ERP. This projection maps the viewing sphere onto the panorama through the longitude and the latitude values. The ERP is extremely popular since it has a straightforward visualisation and it can be easily manipulated with existing editing tools. However, the main drawback is the redundancy of data and the distortion introduced in the pole area. Irregular regions of the sphere are mapped in regular squares on the plane image. For instance, the pole area has to be stretched horizontally in order to fit with the regular area on the plane. This introduces artefacts at the pole in the projected picture. To better understand how this affect the final QoE, Figure 2.6 shows the projection on the planar domain of three selected viewports with different values of elevation. It can be noticed that the viewport based on center position is projected into different regions (stretched or deformed) on the panorama. This results in minimal deformation for the viewports that are at the equator, and large deformation for the viewports with high elevation (*i.e.*, at the poles). Indeed from Figure 2.6, we can notice that the first viewport with $\phi_p = \frac{\pi}{2}$ is the most affected by the projection: the spherical portion has to be stretched to cover the entire corresponding planar area.

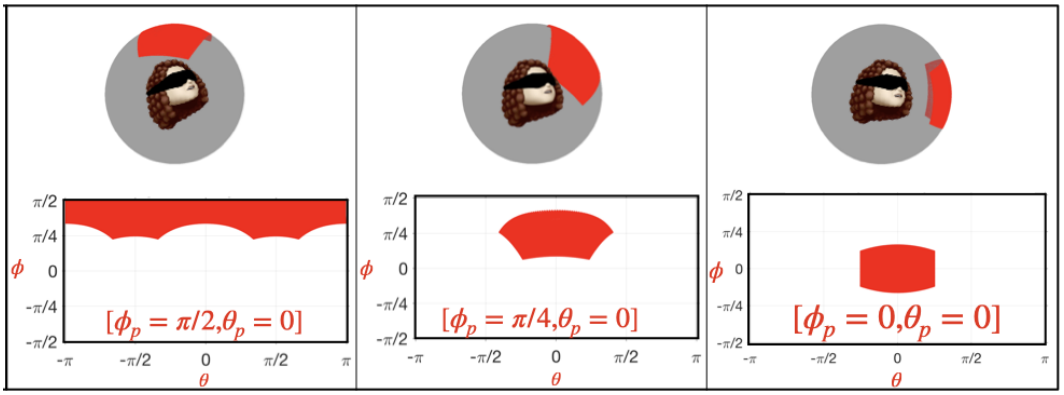


Figure 2.6: From spherical to planar domain by equirectangular projection: comparison of viewport projection from different positions on the sphere. Red shared areas represent viewports.

Cube Map Projection Industries, especially related to gaming, have then introduced a new mapping scheme: the Cube Map Projection. With this projection, the sphere is firstly mapped to a cube. Then, the cube is opened and each of the six faces is arranged into the panorama frame. In this way, no distortion and redundancy is introduced within the cube's face since the sampling on the sphere is regular. However, the unfolding of the cube still compromises the quality because the distribution of pixels increases towards the corners of the cube, as depicted in Figure 2.5 (c).

Pyramid Projection The pyramid projection is an example of variable quality projection proposed recently by Facebook [62]. The sphere is projected on a pyramid with an equal distribution of data points. In particular, the base results to be the privileged faces in terms of quality. However, this projection is still very complex in terms of rendering and it requires huge storage so not really compatible with existing platforms other than have relevant quality differences between faces (base versus lateral faces).

Once the content is projected into a 2D plane, it can be processed by the *Encoding* step in Figure 2.3, using the state-of-the-art codec from classical 2D media compression, such as High-Efficiency Video Coding (HEVC) [63]. The increasing interest in 360° video streaming has risen the urgency to standardise omnidirectional multimedia content [64]. The Omnidirectional Media Application Format (OMAF), included in MPEG-I, specifies the streaming strategies and metadata compatible with DASH. Both equirectangular and cube map projection are supported.

Volumetric Content

Volumetric or hologram content is an emerging media format, still at its infant stage, which has recently attracted a lot of attention due to its ability to represent 3D space

(a) Point Cloud in CWIPC-SXR database [65] **(b)** Mesh from V-SENSE [66]

Figure 2.7: Examples of volumetric representations.

and objects in a realistic manner. As mentioned in Section 2.4, this content can be displayed from any prospective enabling therefore immersive navigation in a 6-DoF VR environment. Volumetric format is typically acquired through a specific configurations of multiple cameras with depth sensors (*e.g.*, Microsoft Kinect, Intel RealSense) or by using LiDAR-based cameras [67]. The acquired volumetric signals can be represented as dynamic *point clouds* or *polygon meshes*. The former is a collection of independent points, defined by their 3D coordinates (x, y, z position) and multiple attributes (*e.g.*, colour, opacity, reflectance, texture) while the later defines 3D shapes as a set of vertices, edges with connectivity information, and textured faces. Examples of these representations are shown in Figure 2.7.
3D meshes have been deeply investigated by computer graphics community because of their ability to accurately represent objects [68]. However, mesh-based solutions turn out to be more complex to manage due to their need to preserve a fixed structure. Point clouds, on the other hand, are simpler and more flexible to acquire and store, and moreover, they outperform meshes in terms of quality at low bit rates [69]. Due to their advantages, point clouds have recently become the most popular volumetric format for real-time applications [70]. However, this type of content typically requires a vast amount of data showing the need for efficient data representations and compression algorithms. For instance, to reduce the amount of data that has to be transmitted while ensuring a good quality level, research efforts have mainly focused on point clouds compression, included an emerging standardi-

sation process [71, 72]. Novel Point Cloud Compression (PCC) have been proposed which can be classified as Video-based PCC (V-PCC), mainly suitable for dynamic point clouds, and Geometry-based PCC (G-PCC), for scenes and objects [73]. The former is based on 3D-to-2D projections so that can take advantage of state-of-the-art traditional video compression techniques (*e.g.*, HEVC); while the later works directly in the 3D space through 3D model data (*e.g.*, octree or triangle surface).

2.2.2 Delivery

After being compressed, the immersive content is almost ready to be delivered to the user over internet network. Nowadays, the dominant standard for video streaming is Dynamic Adaptive Streaming over HTTP (DASH). This protocol offers to the users the possibility to adaptively select different representations of the content, (i.e., different coding rates and resolutions) and typically clients *pull* the content from the server, instead of being the server *pushing* it to clients. Specifically, each content is encoded into multiple resolutions and quality levels (*representations*), and each client dynamically selects the best representations when fetching video segments using HTTP requests. Hence, the encoding step produces multiple quality levels (representations). Each encoded representation is then segmented: the *Segmenting* step breaks the video into *temporal* chunks (usually $2s$ long) and stored at the server side. The different available representations of these chunks are described at the server, which in the case of DASH, is done on through Media Presentation Description (MPD). The client then selects the most appropriate chunk representation to request, as explained later. DASH streaming was extended to the tile-based encoding that has played a key role in viewport dependent streaming. The video content is *spatially* cropped into different bitstreams named tiles, each of those encoded at a different coding rates and resolutions independently from the other tiles. This enables per-tile representations that are stored at the server [56], providing the client the freedom to select unequal quality in the immersive content. The chunks created by the Segmenting step are then ingested in a HTTP origin server that will process clients requests, *Delivery* step. This origin server is usually inside a Content Delivery Network (CDN), a network of connected servers geographically spread in different locations. The CDN organises the delivery of data from the *origin server*, where the content is stored, to the *edge server*, which is the closest server to the client, selecting the quickest and safest route. The transport of high-rate content as VR has posed novel open challenges for the CDN, such as optimal caching, edge computation, etc. However, they are beyond the scope of this thesis, which is addressing mainly to the challenges at the client and server side for immersive media applications.

(a) Oculus Rift [74]. **(b)** Google Cardboard [75].

Figure 2.8: Examples of Head Mounted Display (HMD).

2.2.3 Interactive users

Once the end-user has received the video content, the chunks are processed by *Segment Decapsulation* to extract HEVC bitstream, which is then decoded in the *Decoding* step and fed into the playout buffer. In parallel, still at the decoder side, the *Adaptation Logic* dynamically decides the best representations to request for the upcoming chunk. This selection is based on the client connection and buffer condition as well as the device capabilities. As already mentioned, each media client selects the most appropriate representation of the video content to maximises the experienced video quality and yet meets the network constraints (*e.g.*, estimated bandwidth, buffer level). For example, high quality representations are selected in the case of large bandwidth. Conversely, low quality representations are downloaded in the case of poor (limited) network resources.

The pipeline ends with *Rendering*, which back-projects the decoded planar representation in the spherical geometry and displays the content of interest to the final user. This content of interest (viewport) is evaluated in the prior step *Viewport extraction*, in which the current user viewing direction is translated into the displayed viewport. To display VR video content, different type of devices can be used such as laptop, tablet and smartphone. However, the device that provides a real VR experience is the Head-Mounted Display (HMD): an helmet with a display and movement sensor able to adapt the rendered image to user's head position. Figure 2.8 depicts two examples of popular HMD: the success of VR applications is also demonstrated by the increasing availability of headsets in terms of price.

It is clear that the intelligence of the streaming protocol has been moved to the client side, leading to highly scalable and successful video streaming platforms in Video on Demand (VoD) applications. Netflix and YouTube are only two of the most popular examples. However, due to the huge volume of data that need to be transmitted in the case of immersive content and the unknown clients behaviour,

new challenges need to be faced when extending streaming system to immersive VR applications. In particular, interactive users consume only a portion of the entire content (i.e., the viewport). Sending only the viewport of interest to the user would save bandwidth substantially. At the same time, it is also important to achieve a low streaming delay in order to offer a proper interactive service. For example, the displayed viewport need to be quickly adapt to the head direction. Sending selective viewports cannot address this low-delay requirement, as users dynamically move their head. A possible solution is to send the entire content, minimising the delay but at the price of high data rate transmission. Therefore, it is essential to seek the correct streaming strategy able to find an optimal trade-off between bandwidth efficiency, quality and latency. A brief overview of the research efforts that have been dedicated to advance 3-DoF (*i.e.*, spherical content) and 6-DoF (*i.e.*, volumetric content) VR streaming strategies is given in the following in Section 2.3 and 2.4, respectively.

2.3 System-centric approaches for 3-DoF VR

Advances in 3-DoF VR streaming were initially aimed at improving the overall system performance in terms of consumed bandwidth, storage cost, and networking reliability metrics. Therefore, we name them as *system-centric streaming solutions*, being usually user-agnostic (neglecting user behaviour analyses and prediction) and being instead system-aware. In the following, we further categorise system-centric solutions for ODV in viewport-independent and viewport-dependent streaming. *Viewport-independent* solutions are the ones most similar to traditional adaptive 2D video streaming, in which the entire panorama is encoded and treated equally. Specifically, each representation available at the server side is encoded with a uniform quality and resolution across the entire panorama, as shown in the first row of the content provider box in Figure 2.9. Then, any final client is able to select the representation that best fits their networking and displaying requirements. This ensures zero latency for viewport-switching, but it is extremely costly in terms of storage and bandwidth usage [76]. Moreover, it assumes that the whole panorama is equally important, which is clearly not the case in VR systems. In fact, only a portion of the panorama is actually transmitted, meaning that not all content is needed at the client side, despite all being sent. To strike the optimal balance between bandwidth waste and switching latency, ODV user interactivity needs to be taken into account, leading to streaming strategies adapting not only to the content but also to users, *viewport-dependent* streaming. The key intuition is that each step of the streaming pipeline should prioritise the content that is most likely displayed by the interactive users, as initially shown in 2012 by Alface *et al.* [77] .

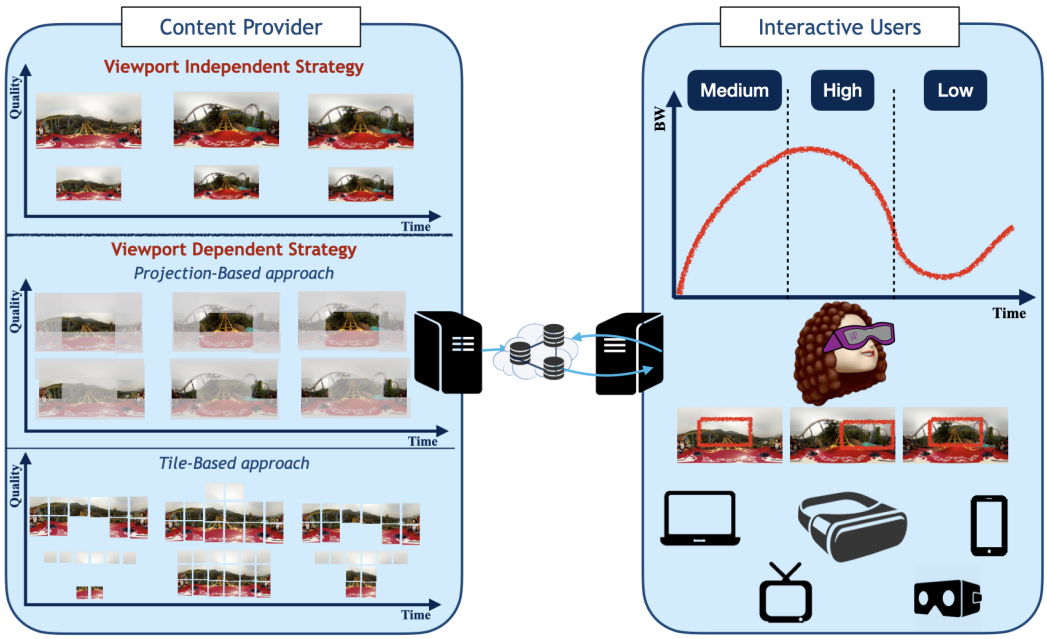


Figure 2.9: Conceptual differences of system-centric approaches for 3-DoF VR: viewport-independent vs. viewport-dependent (projection- and tile-based approaches) strategies.

In the following, we present in details advances on both viewport-independent and -dependent streaming solutions.

2.3.1 Viewport-Independent strategies

Beyond initial works that applied classical DASH streaming to ODVs [44], recent works that fall under the viewport-independent framework are mainly focused on improving the encoding step, mainly overcoming issues related to sphere-to-plane projection. One goal is to encode the ODV directly in the spherical domain with no need to project the content into a bi-dimensional space. For instance, Vishwanath *et al.* [78] propose to encode the inter-frame motion vector (MV) directly on the sphere and not on the planar domain, avoiding distortions during the sphere-to-plane conversion. Also working on the sphere, Bidgoli *et al.* [79] propose a representation learning approach for spherical data. To avoid distortion from planar projection, the authors redefine classical convolution operation for 2D images on the spherical domain exploiting their low complexity while having the advantage of ensuring an uniform sampling on the sphere. This can lead to better coding efficiency, as proved in the paper. These aforementioned approaches show promising usage of on-the-sphere encoding strategies for omnidirectional images.

2.3.2 Viewport-Dependent strategies

The key intuition to strike the optimal balance between bandwidth waste and switching latency is that the content is not equally important (*spatially*), since users display

only a portion of it. Therefore, it is clear that a non-uniform streaming solutions, ensuring higher quality to the areas more likely to be displayed, should be considered. In this direction, the viewport-dependent strategy is a viable solution, in which the panorama is encoded into one single representation, but with non-uniform quality across the panorama. Each representation encodes the entire panorama, with one region at higher quality and the remaining encoded at lower quality. Different representations have different high quality regions. Then, each user selects the representation that better fits their viewing direction and network resources. The adaptation to the viewport has been implemented in both the projection and the (tile-based) encoding step of the streaming pipeline. The former (*projection-based approach*) adapts the bit rate allocation during projection, making this an unequal allocation that prioritises areas of interest for the final users (central row of the content provider box in Figure 2.9). The latter (*tile-based approach*) has emerged when tiled based coding was proposed and most of the ODV works have focused afterwards on this streaming technique (last row of the content provider box in Figure 2.9). In the following, we detail these two approaches.

Projection-based approach

Projection-based approaches aim at projecting the spherical content in such a way that the most important areas (most likely to be displayed) are the least distorted during the projection step of the pipeline. This unequal level of encoding rates across the panorama offers better quality experienced by the users, but at a price of generating multiple versions of the same panorama (each version with a different area at high-quality). This impacts the storage cost (higher than the viewport-independent case) and other systems metrics such as the caching hit ratio (lower since users might prefer different versions depending on their focus of attention). The first example of this strategy has been presented by Facebook Inc. in [80] where different representations are obtained by pyramid projections (Figure 2.5). As introduced in Section 2.2.1, this novel mapping projects the sphere onto a pyramid, with an intentional unequal allocation of points – more data points to the pyramid base. The pyramid is then unwrapped and resized into 2D space and encoded with classical 2D codec tools. The novelty is that this projection encodes at the higher quality the area corresponding to the bottom face of the pyramid, which will be match with the viewing direction of the users. The projection however suffers from a rapid quality degradation when moving from the bottom face to the lateral ones. Then, authors in [81] have presented a similar solution but based on Truncated Square Pyramid (TSP) projection. This projection comes from the traditional one but has less quality degradation between the bottom and the other faces.

Tile-based approach

An alternative, viewport-adaptive streaming has been presented in [82]. The key intuition is that each representation is characterised by a specific Region of Interest (RoI) in the scene. This portion is encoded at higher quality while the remaining part at a quality that is decreasing with the distance from the centre of this area, named Quality Emphasis Centre (QEC). Then, the client downloads the version with the QEC closer to his viewing direction. As main novelty, the aforementioned pillar work [82] has introduced the concept that the content is not spatially equally important, opening the gate to tile-based approach. Today, it has been advanced by current standardised tiled streaming, which encodes tiles at different bitrates and resolutions, creating per-tile representations that are stored at the server. The key novelty of a tile-based system is that tiles can be independently fetched by the client to compose the viewport at the desired quality level. This creates a more flexible transmission strategy able to increase the quality only of the small portion that will be displayed. However, this come at the price of a reduced coding efficiency since the encoding is limited to the tiles. In fact tiles are smaller than the entire panorama and then, have less redundancy to exploit during the coding. Moreover, more complexity is added at the client that has first to select the more suitable tiles of the panorama and, once received, fetched them together and finally extract the currently viewport. The drawbacks notwithstanding, the adoption of tile-based encoding has been definitely justified by the gain in flexibility (in terms of users that can be served) [77]. For this reason, tile-based encoding has become the de-facto encoding strategy for 3-DoF VR. In the following, we describe the main recent works focused on 1) optimal design of the tiles at the encoder side; 2) optimised *adaptation logic* (*i.e.*, strategy to request the desired content) at the client side that seek to find the optimal trade-off between bandwidth efficiency, quality and latency in a tile-based system.

The tile-based encoding strategy directly impacts the server-side storage costs, hence the compelling need to optimise it for omnidirectional content. Similarly to the projection-based solutions, a limited number of representations will have to be encoded to control the storage costs, at the price of reducing the flexibility of the algorithms. To ensure the right flexibility (*i.e.*, to be able to serve as much as heterogeneous users as possible), many representations need to be encoded, with different high-quality regions and with different encoding-resolution parameters. It is worth mentioning that encoding and storing have a cost for the server or for the service provider. It is therefore important to optimise properly the server storage space.

Works focused on this optimisation for classical adaptive streaming platforms have been already proposed in the literature [83], tuning the coding rate and resolution depending on both the users' behaviour and the type of content. In the case of omnidirectional video content, however, there is an extra degree of freedom that is the high-quality area definition. If before an entire chunk was encoded with given coding parameters, now different representations from the same video content can be encoded at different rates and resolutions depending on the users' interest. In this context, [84] introduces a new content-aware encoding ladder estimation method for 360° VR video in adaptive streaming system. The aim of this work is to find the optimal coding parameters that optimises the QoE perceived by the final users while minimising the storage and coding costs.

Instead of optimising the tile design at the server side, many other works have been focused on optimising designs at the user side, studying optimal adaptation logic in viewport-dependent tile-based solutions. Ashan *et al.* [85] propose a buffer-based algorithm to optimize the bitrate of area in the panorama overlapping the users viewport. The main challenge is that this does not simply imply selecting the optimal representation from the MPD, but it rather needs a mapping from tiles bitrate to viewport quality. Other efforts instead focused on adapting the strategy to users interaction, exploiting either the knowledge of the viewing direction [86] or probabilistic/average models such as the heatmap [87]. Fu *et al.* [86] propose an adaptation logic based on sequential reinforcement learning (RL), with the agent reward being the quality experienced by the users. The RL agent implicitly learns the interaction model and refines the adaptation logic based on the users behaviour, showing a gain of 12% QoE improvement with respect to non viewport-dependent strategies. Ozcinar *et al.* [87] propose a visual attention-based ODV streaming system optimising the tile-based design taking into account users heatmap. Other parallel researches have improved network aspects such as multi-path, pre-fetching, decoding offloading, and caching strategies.

The works mentioned so far have shown the gain of proposing viewport-dependent tile-based streaming and its impact across different steps of the pipeline. However, limitations still remain in the intrinsic definition of tile-based streaming for instance related with the reduced encoding efficiency. To obviate this limitation, Son *et al.* [88] propose new HEVC extensions to enable inter-frame prediction, when an encoded object has dependency from outside the tile using the HEVC scalability extension and extracting the object from an up-sampled version of the base layer. Another limitation of tile-based coding is the intra-frame redundancy, which is not minimized across tiles. Bidgoli *et al.* [89] address this limitation by allowing the encoder to detect reference regions outside the current fetched tiles. Then, they

propose an encoding in which the intra-frame reference region is at the centre of a requested tile to improve client intra-frame decoding.

Despite these last limitations, viewport-dependent tile-based streaming has been widely adopted for ODV streaming. A key aspect across all these works is the user's information, expressed for example either via viewport trajectory or heatmap. The advances made so far have motivated researchers to dig deeper in the study of user's behaviour and leading to personalised systems, which put the user at the center of the system and tailor every aspect of the coding–delivery–rendering chain to the viewer interaction (*i.e.*, *user-centric streaming*). As main contribution of this thesis, Chapter 3 overviews how the role of the interactive user have been analysed in the literature and describes how such interactivity can drastically improve the status quo using user-centric streaming in 3-DoF VR streaming system.

2.4 Emerging 6-DoF VR streaming systems

An increasing research interest has been showed for the streaming of volumetric content, the de facto multimedia format for emerging 6-DoF VR systems. Even if volumetric signal has a completely different structure (a collection of unstructured colour points information evolving over time as describe in Section 2.2.1) from spherical content, they both share some common challenges with spherical video such as a large volume of data, ultra-low delay application, users display only a restricted portion (*i.e.*, viewport), and consequently their uncertainty in navigating within the content. Thus, most of the studies presented so far to improve and make real immersive systems for volumetric content are straightforward extensions of traditional solutions for ODVs. The first attempts of point cloud streaming systems are based on tiles, similarly to ODV solutions [90, 91]. In detail, Park *et al.* [90] split the point cloud into smaller equal 3D cuboids, and each of them correspond to one tile. They define an utility function to optimise the set of requested tiles taking into account both point cloud features, such as data rate and decoding complexity, and users' distance from the content. In their preliminary work, Hosseini *et al.* [91] instead extend adaptive streaming to point cloud proposing a view-aware system named DASH-PC and a specific MPD for volumetric content. Also their approach is based on tiles to enable different quality representations for the displayed portion, and thus save bandwidth. On this direction, Subramanyam *et al.* [92] propose a low-complexity approach in which point clouds are divided into non-overlapped vertical slices to be independently decodable. Then, the system allocates quality rate for each tiles depending on user's position on the virtual floor and viewing direction. Authors have tested their proposed system on real navigation trajectories showing a considerable bit-rate gains in contrast with non-adaptive strategies. Due

to occlusion, typically half points of the point cloud are not visible from any user viewing position. To overcome this issue, He *et al.* [93] consider a view-dependent streaming system in which the volumetric content is divided into six faces using a cubic projection. Each face is then compressed with traditional 2D compression techniques and delivered to the user on a hybrid (*i.e.*, broadband and broadcast) networks. Recently, a growing attention has been put also to enable the deliver of volumetric content to mobile devices (*i.e.*, smartphones) [94, 95]. Nebula is a holistic mobile system which reduce the spatial density of point cloud on the edge server in order to save bandwidth and minimise computational complexity at the client side [94]. ViVo is a recent practical framework based on visibility-aware optimisation scheme [95]. These preliminary works, however, consider only a single point clouds in the scene to be delivered. A step forward in this direction has been presented in [96]. Hooft *et al.* indeed propose and compare different rate adaptation schemes, named as PCC DASH, for adaptive streaming of multiple dynamic point cloud objects. In particular, these rate adaptation heuristics take into account user's position, viewing direction, content information (*i.e.*, available representations, positions of each object in the scene) and system information (*i.e.*, available bandwidth, buffer status).

In this first part of the thesis, the main relevant background knowledge related with immersive communication system have been introduced. This poses the basis for establishing the importance and for better understanding the contributions done in this PhD thesis.

Part II

BEHAVIOURAL ANALYSIS IN 3-DoF VR

Chapter 3

The role of the user in 3-DoF VR system

In this part of the thesis, we focus on the role of the user in 3-DoF VR system, and on the importance of deeply understanding users behaviour to enable user centric solutions for the next-generation of multimedia systems. First, we present in this chapter an in-depth overview of the research efforts that have been done to analyse user behaviour and how such information has been leveraged to advance ODV streaming strategies in the latest user-centric systems. Then, we introduce our proposed tools for behavioural analysis of users navigating in a 3-DoF VR system: a novel spherical clustering algorithm to detect behavioural similarities among users (Chapter 4), and information theoretic metrics for quantifying similarities for the same viewer across contents (Chapter 5).

3.1 Introduction

The previous chapter has highlighted that the new format (*i.e.*, spherical) as well as the new way of consuming the content open the gate to many promising VR applications, but also pose completely new challenges. One key challenge raised from the interactivity level is the high resolution and low-latency required to ensure a full sense of presence. The user needs to have ultra-low switching delays when changing the displayed viewport to avoid discomfort. This can be ensured by sending to all users the entire content at high-quality, assuming that the desired viewport will be then exported during the rendering. This solution is the first one that has been proposed, extending the well-established and optimised methods for 2D videos to spherical content. These methodologies are usually user-agnostic and aimed at improving the overall performance through enhancements in the system, therefore we have named them as *system-centric streaming*. The research efforts done following this line of improvements has been already presented in Section 2.3. The main

Table 3.1: Surveys related with ODVs streaming systems. Level of investigation per each topic: ■ mentioned; ■■ sufficient; ■■■ deep.

Survey	Content Preparation	Compression	Delivery	Rendering	Quality Assessment	Prediction	Behavioural Analysis
Chen *et al.* [99]		■■■	■■		■■■		
He *et al.* [100]		■■	■■			■■	
Fan *et al.* [101]	■■	■■■	■■■		■■		
Zink *et al.* [102]	■	■	■	■	■		
Azevedo *et al.* [103]	■■■	■■	■	■■■	■■		
Yaqoob *et al.* [104]		■■	■■■		■■	■■	
Shafi *et al.* [105]		■■■	■■■	■■	■■■		
Xu *et al.* [106]		■■■	■■		■■■	■■	■
Ruan *et al.* [107]	■	■	■	■			
Chiarotti [108]		■■	■■■		■■■	■■■	

drawback of these solutions is that they are extremely bandwidth-consuming (up to 43 MB/s for 8K spherical video [97]) since the entire ODV content is delivered to final clients, pushing the available bandwidth to the limit, with a negative impact on the final quality. In practice, only a small portion (typically around 15% of the entire video [98]) of the overall content is displayed by the user, making these solutions extremely inefficient. Recently, more attention has been put on the final users, leading to personalised systems, which put the user at the center of the system and tailor every aspect of the coding–delivery–rendering chain to the viewer interaction. For example, only the predicted content of interest for the final user is pushed into the delivery network. The main aim of these personalised systems is to optimise the user QoE but also to overcome ODVs streaming limitations, such as reducing bandwidth and storage usage. However, this comes at the price of requiring the knowledge of user interactivity patterns in advance. Thus, we name 3-DoF VR system that follows this second approach for improvements as *user-centric streaming*.

In this chapter, we provide an overview of the research efforts that have been dedicated to advance ODV streaming strategies, with a specific attention to the more recent user-centric systems. Due to this popularity, many surveys papers [99–108] have been published to summarise the main contributions to ODVs streaming systems. Table 3.1 depicts these works visualising their main topics of interest, highlighting also the level of investigation across the end-to-end pipeline. As it is evident from Table 3.1, the majority of existing surveys are deeply focused on compression, delivery and quality assessment aspects. For instance, Zink *et al.* [102] have provided a general overview of the main challenges and the first attempts of solution per each step of the ODV streaming pipeline, from acquisition to the final user rendering experience. Chen *et al.* [99] have mainly explored the most re-

cent projections methods aimed at improving video coding and transmissions, and reducing video quality distortions. More insights on system design and implementations have been described in [101, 104, 107]. In particular, both Fan *et al.* [101] and Yaqoob *et al.* [104] have examined existing protocols and standards together with optimal ODVs streaming solutions. Ruan *et al.* [107] have instead investigated solutions for VR systems but mainly from a network services perspective. Visual quality artefacts have been deeply investigated by Azevedo *et al.* [103] describing their sources and features at each step of the system; authors have also presented an overview of existing tools for quality assessment (objective and subjective). Similarly, a deep focus on visual quality assessment, together with attention models and compression, is given by Xu *et al.* [106]. Authors have highlighted the importance of predicting where viewers mainly put their attention during immersive navigation (*i.e.*, saliency maps) to benefit the entire system since users are the final consumers. They have also partially addressed the need of understanding behavioural features to help in modelling user attention presenting the main outcomes from existing navigation dataset analysis. Following this direction, the recent work presented by Chiarotti [108] has showed the importance of estimating navigation path also for quality evaluation, neglecting however the behavioural analysis. To the best of our knowledge, these are the only existing surveys which explicitly brings out the importance of the new role of users in ODV streaming applications, and thus the need of understanding their behaviour. As shown in Table 3.1, behavioural analysis has been highly overlooked. One of the main contributions of this chapter is to fill in this gap by discussing in-depth the role of the user in ODV streaming strategies.

The reaming of this chapter is structured as follows: Section 3.2 provides an overview on the recent ODV multimedia datasets currently available to our community while Section 3.3 describes the different approaches used to perform behavioural data analysis within spherical content. Section 3.4 shows how such novel interactivity can drastically improve the status quo using user-centric streaming. To conclude, we summarise the chapter with final remarks in Section 3.5.

3.2 Existing ODV navigation datasets

We provide an overview of the datasets collecting user's navigation data during immersive experiences. We summarise these datasets in Table 3.2, and highlight that they are limited to *i) publicly* available dataset, with data related to *ii)* ODV content (no images), and *iii)* navigation trajectories (*i.e.*, head and/or eye movements). In order to mimic a real-life scenario, VR users cannot display the entire environment around themself but only a restricted portion (i.e., viewport). As defined

Table 3.2: ODV navigation datasets publicly available. Link to each dataset can be found in the Bibliography.

Y	Reference	Test Preparation			Subj.	Available Data	
		ODVs	Len.	Category		Format	Others
2017	Corbillon [109]	5	70*s.*	Content Genres	59	Quaternion	Open source software.
	Lo [110]	10	60*s.*		50	Euler angles	Saliency and motion maps.
	Wu [111]	18	164-655*s.*	Content Genres	48	Quaternion	Free-task and Task experiments.
	Xu, M. [112]	48	20-60*s.*	Content Genres	40	Spherical coord.	VQA task.
2018	Ozcinar [113]	6	10*s.*	Content Genres	17	Spherical coord.	Open source software.
	Fremerey [114]	20	30*s.*	-	48	Euler angles	Open source software.
	Xu, M. [115]	58	10-80*s.*	Content Genres	76	Spherical coord.	HM and EM data.
	David [116]	19	20*s.*	Content Features	57	Spherical coord.	HM and EM data, saliency maps.
	Zhang [117]	104	20-60*s.*	Content Genres	20★	Spherical coord.	HM and EM data and heatmaps.
	Xu, Y. [118]	208	20-60*s.*	Content Genres & Feat., Camera Motion	31★	Spherical coord.	HM and EM data.
2019	Nasrabadi [119]	28	60*s.*	Camera Motion	60	Quaternion	Questionnaire on attention.
2020	Rossi [34]	15	20*s.*	Content Genres	31▲	Spherical coord.	Data also from Laptop and Tablet, code ODV storage optimisation.
	Rondón [120]	306	20s-655s	-	∼42★	Spherical coord.	Aggregation datasets [109–111, 115, 116, 118].
2021	Dharmasiri [121]	88	30-655*s.*	-	∼45★	Euler angles ♦	Aggregation datasets [109–111, 119, 122, 123], code video segment categorization.
	Chakareski [124]	15	36*s.*	Content Genres	5-12	Euler angles	RD characteristics of full UHD ODVs.

★ per video.
▲ per video and device.
♦ roll angle was ignored.

in Section 2.1, the *navigation trajectories* identify the movements of users while experiencing an immersive content. Specifically, the sequence of spatio-temporal points representing the user's viewing direction over time identifies users navigation within an immersive experience. Based on VR device technology, the user's viewing direction can be represented either by head or eye movements. The head movement determines the FoV as the pixel area of ODV, which is displayed by a given user over time, while the eye movement datasets contain the specific area within the FoV that captures the user attention and can be classified as salient.

The navigation trajectories are collected via a three-steps collection procedure: *1)* Test Preparation, *2)* Subjective Test, and *3)* Data Formatting and Storage. As first step researchers select the video content to use during their subjective experiments. During *test preparation*, ODVs are selected based on several criteria: video length, number of video, content category, features and attributes. Considering the video sequence length, Ozcinar *et al.* present in [113] the shortest group of video (*i.e.*, 10 sec.) while Duanm *et al.* [125] have the longest one with sequence in the range of 60-120 seconds. The widest range of ODVs is instead proposed by Xu *et al.* in [118] with videos of variable length, from 10 up to 80 seconds. Also, in terms of number of video contents there are various choices: only 5 videos (with 2 more ODVs used during the training phase) in [109] as opposed to 208 ODVs in [118]. Interestingly, two of the most recent dataset presented [120, 121] integrate previous databases such as [109–111, 115, 116, 118] and [109, 110, 119, 122, 123], respectively. Therefore, they become among the largest and most heterogeneous ODV datasets currently publicly available. Another criteria to select ODVs is based on three main categories: *Content Genres*, *Content Features*, and *Camera Motion*. For instance, authors in [117] select their video only based on the content Genres, offering mainly sport activities related videos. A wider range of genres (*e.g.*, music shows, documentaries, short movies, computer animation and gaming) can be found in most of the publicly available datasets [34, 111, 112, 115]. Other authors choose their content based on attributes such as camera motion for [119], outdoor/indoor scene in [116] or a mix of all the aforementioned video categories (*i.e.*, indoor/outdoor scene, fixed/moving camera, and different content genres) in [118]. It is worth to mentioning that the most recent dataset presented by Chakareski *et al.* [124] is the only one which presents full Ultra High Definition (UHD) ODVs.

The second step of the data collection campaign is the *subjective test*, which represents the core data collection step. Most of the datasets collect navigation data during free-task experiments, which means that viewers could move inside the video content as they wished. There are, however, a few examples where users

were asked to take some specific actions. For instance, the work presented in [111] proposes two different experiments: a first set of ODVs are used to identify the natural behaviour in a free-navigation experiment, while a second one is more specific to VR live streaming applications. In fact, in the second experiment live recorded ODV have been used to mimic the case of live-streaming data-tracking, and beyond objective head-movements trajectories, also subjective perception of the video content was captured. Similarly, authors in [119] study participant's attention, presence and discomfort levels by a questionnaire at the end of the vision session. Auxiliary subjective quality scores were collected also in [112], in which the dataset is used for Visual Quality Assessment (VQA) tasks. Looking more specifically at the capturing of the objective data (*i.e.*, trajectories), there are different types of VR devices that can be used, such as laptop, tablet and smartphone. Under the assumption that it provides to most immerse experience, the most widely adopted device is the the Head-Mounted Display (HMD). Other datasets however do exist with data collected by other devices. Duanmu *et al.* [125] propose navigation trajectories experienced only on laptop. In this context, in Chapter 4 we present a collection of users trajectories across multiple devices (tablet and laptop, in addition to HMD) with the main purpose of studying the effect of the displaying device in the user's navigation. Moreover, most of the presented ODV datasets provide the viewers navigation trajectories as a sequence of head movements over time [109–115]. Even if the head position is a valuable proxy of the user viewing direction, people can still move their eyes and focus on a specific area of the displayed viewport keeping the head fixed. Thus for specific applications, such as visual attention modelling or quality assessment, recording eye gaze movements during the navigation is equally valuable. Hence, there are also dataset containing both information [116–118].

The last part of the creation of an ODV dataset is the *data formatting and storage* of the collected navigation trajectories in an immersive scenario. Since the navigation within ODV is restricted to 3-DoF movements, only rotational movements are captured neglecting potential translational movements. These rotational movements can be represented based on several conventions within a spherical system: Euler angles (*i.e.*, yaw, pitch and roll), spherical coordinates (*i.e.*, latitude and longitude), and quaternion. The first two formats are the most common, as shown in Table 3.2, while quaternion is employed only by [109, 119], highlighting the higher accuracy and robustness of the quaternion in representing rotational movements. Finally, some of the current publicly available dataset provide also some other data: software that have been used to record users navigation during subjective experiments in order to encourage the community to extend their collected data [109, 113, 114]; saliency [116] and motion maps [110]; other algorithms such

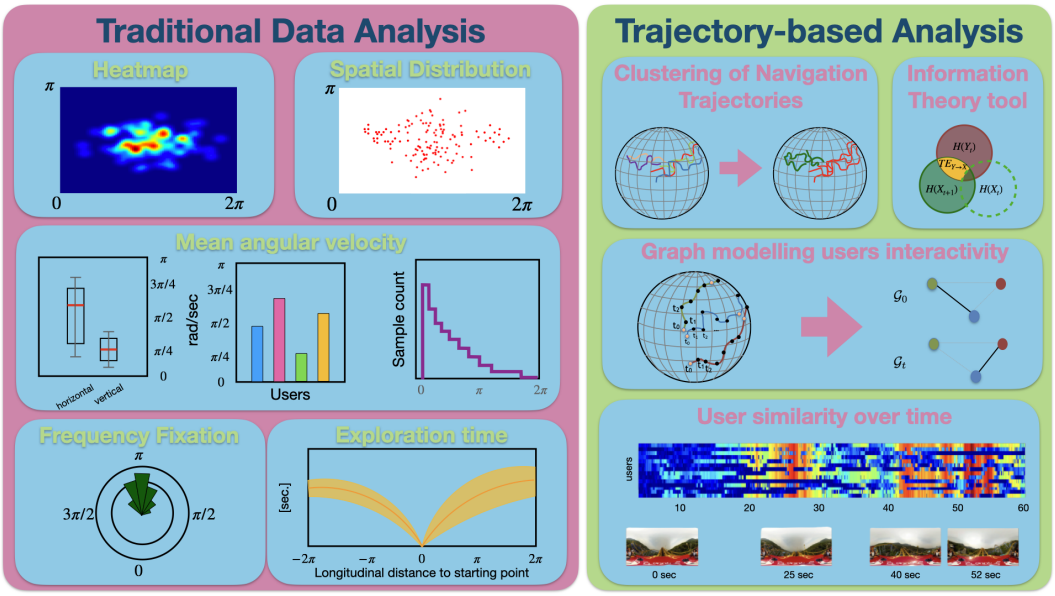


Figure 3.1: Taxonomy of behavioural data analysis in VR.

server storage optimisation and video segment categorization in [34] and [121], respectively; Rate-Distortion (RD) characteristics of UHD ODV to correlate with user navigation in [124].

3.3 Behavioural Analysis within ODVs

Most of the works presented in the previous section provide, along with the dataset, a general statistical characterisation of users behaviour. This has opened the gate to a new research area aimed at capturing key features that characterise users interaction while experiencing ODV. In the following, we depict the main findings of this prominent area of research on behavioural analysis. In particular, we distinguish two strands of investigation: a more traditional one aimed at identifying general behavioural features of users while navigating; and a second one focused on identifying more specific and representative users features of the navigation behaviour such as users similarity based on trajectory-based data analysis. Figure 3.1 provides visual examples of key metrics used for both these line of investigations.

3.3.1 Traditional Data Analysis

The way in which users typically interact with ODVs has been analysed mainly in terms of general metrics such as angular velocity, frequency of fixation, and mean exploration angles. Other than these quantitative metrics, a visual (and qualitative) tool used to study user's behaviour in VR is the heatmap, which identifies areas of the content mostly attended by viewers within a time interval. The investigations based on these aforementioned metrics have given intuitions to answer some key-questions. Finding an answer for these issues is indeed a first step towards

the design of user-centric solutions for ODV system. We now summarise the most relevant questions (from a more generic to a more specific behavioural perspective) and the works that aimed at answering them.

Where do users usually look at (on average)? Understanding the areas on which users focus the most within the spherical content is key to optimise the streaming pipeline via viewport-dependent coding techniques or adaptation logic as described in Section 2.3.2, and thus ensuring a good final quality. For this reason, many different researches at first focused on showing through statistical analysis that users prefer to look at the equatorial area of ODVs rather than at the poles [109, 113, 116–118]. For instance, [109, 113] use heatmaps averaged across users per each content to show that viewers head is densely distributed over time in the equatorial region and in particular, above 100° in terms of latitude. A similar behaviour has been confirmed by Xu *et al.* [118] analysing eye movements: the distribution of gaze fixations shows indeed an equatorial tendency, and moreover users move more frequently on the left and right directions than up and down. Deeper investigations on the equatorial bias have highlighted that viewers spend more time towards the front center area of ODVs where typical the main relevant scene is located [112, 114, 115, 125]. Finally, Fremerey *et al.* [114] illustrate in their behavioural analysis that users change their viewing direction only for a short period of time and in general prefer to display the video from a more central and comfortable position.

How do users actually move over time? The average spatial distribution of users attention might not be enough to characterise their behaviour. For example, a deeper understanding on how viewers actually navigate over time within the video content would allow us to distinguish behaviours that can be predicted or not. In detail, erratic and random navigation within the spherical content with not consistency (across content or users) is usually challenging to predict. This is the case of the beginning of the navigation within new ODV, in which a predominance of exploratory movements has been pointed out highlighting their randomness, and therefore their difficulty to be anticipated [109, 111, 121]. However, after a period in the range of 10-20 seconds, viewers tend to converge to common (and more predictable) directions, which typically correspond to the main Focus of Attention (FoA) in the scene. Once users find their main point of interest, they tend to not move too much [121]. Corbillon *et al.* [109] show that in a window of 2 seconds, the 95% of analysed subjects stay within a ray of $\pi/2$ from the initial position. This more static and understandable (FoA driven) behavior make the users interaction

much more predictable than the initial explorative phase.

How is the user behaviour affected by VR devices? Since ODVs can be experienced by different apparatus, a consequent rising question is on the influence of the selected VR devices on the users interactivity. Focusing on the spatial average distribution, the device seems to have a small impact on the overall visitation density: a central and equatorial bias is preserved when navigating via a desktop [125]. Looking at the users navigation over time, however, a dependency on the device has been observed. After a highly-exploratory behaviour at the beginning of the immersive experience (which remains despite the adopted viewing device [126]), a more dynamic navigation has been observed for users displaying ODVs with a laptop than with HMD or tablet, as shown in Chapter 4. For example, this can be related to a lower sense of immersion and engagement that leads to more explorative movements. Thus, taking into account different VR devices is relevant and fruitful to understand the difference in terms of user interactivity. Further analysis has been presented by Broeck *et al.* [126], focusing on user experience with heterogeneous VR devices such as HMD, tablet and smartphones. In particular, two kinds of omnidirectional video sequences were analysed: video with a static viewport (*e.g.*, recorded with a fixed camera) and moving viewport. The results show that users explore more in the latter type of content. As expected, the immersion sensation is higher with HMD than other devices while the tablet offers less immersion. Beyond traditional objective (implicit) metrics such as statistical tool, VR experience can be analysed through subjective (explicit) metrics such as users feedback, and interesting observations can be deduced by comparing implicit and explicit feedback. In [126], authors interviewed participants querying about their sense of immersion during the experience. This explicit data matches the more implicit ones based on users movements since subjects felt more immerse with HMD, despite being less comfortable than with smartphones.

How is the user behaviour affected by content genres? Similarly to the previous question, the correlation between users movements and video content has been analysed from different perspectives. The general behaviour of viewers to be more inclined to display ODVs from a central position might be affected by the video content; for this reason, the correlation between user's movements and video content has been analysed from different perspectives. For instance, Xu M. *et al.* [112] provide a few examples in which the main objects in the scene are not located in the central area of the video and thus, people are more focused on a different area than the central one. Ozcinar *et al.* [113] show a direct correlation

between the distribution of fixation points and the video complexity, in terms of Spatial Information (SI) and Temporal Information (TI). In particular, the lower is the TI, the greater is the number of fixations located in the same region. Moreover, the way in which users navigate inside ODVs change for different video categories [34, 127]. For instance, Almquist *et al.* [127] highlight that viewers tend to be more uniformly distributed for videos without moving objects. In Chapter 4, we observe that ODVs without a main object in the scene bring users to have highly exploratory interaction, especially with HMD. However, we also highlight that the level of interactivity is not correlated to the viewing device if there is a main focus of attention in the video which capture user attention. Finally, the correlation between navigation patterns and the video content is so relevant such that Dharmasiri *et al.* [121] use the fixation distribution as a proxy for their video categorisation algorithm.

Are users consistent in their navigation? A final and yet essential question is if users tend to navigate in a consistent way. In other words, researchers have studied the average behaviour of users but also their variance and deviation. Small variance means that heatmaps can be representative enough of the users behaviour, leading to reliable prediction of the user interaction. A general high consistency among users in terms of spatial distribution has been identified by general statistical analysis for head movements. For instance, Xu *et al.* [112] evaluate an high linear correlation between heatmaps generated by two random sets of users. A similar outcome has been shown in [115] where in particular, it has been highlighted that almost 50% of users focus on the same viewing area among 8 quantized regions of the ODV. These works mainly focused on summary statics such as mean and variance of users behaviour averaged across users. Looking at deeper analysis focused on pairwise comparison, authors in [118] have evaluated the average intersection angle of eye-gaze direction per each pair of participants across each content. This analysis highlights heterogeneity in users behaviour, in contrast with the observations carried out from the other studies. This inconsistency suggests the need to go beyond traditional statistical analysis to better understand the user behaviour within immersive content. Moreover, beyond the inconsistency just highlighted, it is worth mention that most of the above studies are focused on behaviours averaged over time – for example heatmaps and eye fixation. These metrics are highly informative about the spatial behaviour (where do users tend to look at) but only partially informative about at the temporal behaviour (we can deduce how much erratic users tend to be, but not really if two users are interacting similarly). However, deeper temporal analysis is essential to develop deeper behavioural analysis,

reliable users prediction, and thus enable user-centric systems. In the following, we then review this second strand of research focused on trajectory-based data analysis and highlight the key outcomes and novelty that emerge and how this can lead to user-centric systems.

3.3.2 Trajectory-Based Data Analysis

As emerged in the previous discussion, behavioural analysis based on statistical tools or heatmap provides a general understanding of user's behaviour in ODV content failing in detecting deep insights into users navigation dynamics, such as how much viewers interact in harmony among themselves. Specifically, there are still crucial questions that need to be addressed: "*Are users interacting in a similar way? Thus, can human behaviour be predicted?*" Answering to these questions is indeed essential for many and not trivial tasks. In this context, detecting viewers who are navigating in a similar way would help to improve the accuracy and robustness of predictive algorithm and thus, to personalise the delivery strategy. This information could also be exploited for identifying key navigation trajectories which can be used either to optimise video content coding or QoE assessment. Beyond immersive video streaming system applications, being able to detect users who are interacting in a similar way from those who are not, might be essential for medical purposes such as studying psychiatric disorders [128, 129]. Equipped with this motivation, a new direction of behavioural analysis has started aimed at identifying behaviour similarities among users, across video content and/or devices. This research direction is also one of the main focus of this thesis. Therefore, in the following we introduce and contextualise our main contributions that will be discussed in the coming chapters.

Clustering is one of the most popular and robust techniques to infer data structure and it has been therefore employed in the context of VR applications. Based on intuitions from vehicle trajectory prediction, Petrangeli *et al.* [130] model each user navigation as independent trajectories in terms of roll, pitch, and yaw angles, and apply a spectral clustering [131, 132] to identify trajectories with similar behaviour over time. The dominant trajectories, identified by the main clusters are eventually used to predict new viewers. While this method is efficient in discovering general trends of users navigation, it is not focused on identifying clusters that are consistent in terms of displayed content; meaning that users in the same group do not necessarily consume the same portion of ODV. Clustering to perform long-term trajectory predictions is presented in [133], where authors first adopt a well-known spectral clustering algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), to identify key clusters and trajectories from a set of training samples.

Then, when new (test) users start an ODV streaming session, they are assigned to a specific cluster and future trajectories are predicted accordingly. This association step is based on viewing direction comparison measured in the equirectangular planar, neglecting the actual spherical geometry. Therefore, the clusters identified by both algorithms suffer of two major shortcomings: *i)* not identifying group of users necessarily consuming the same portion of the ODV; *ii)* not necessarily considering the spherical geometry into account. To overcome these limitations, in Chapter 4 we propose a novel spherical clustering tools specific for ODV users. This is a graph-based algorithm able to identify groups of viewers based on their consistency in the navigation. In practice, we first define a metric to quantify the common displayed portion of the content (*i.e.*, overlapped viewport) among users; based on this metric, we build a graph whose nodes are associated to different viewers. Finally, a clustering method based on Bron-Kerbosh algorithm is applied to build clusters as cliques (*i.e.*, sub-graph of inter-connected nodes). Thus, the algorithm detects and groups users *only if* they consistently display similar viewports over time while consuming the same ODV content, and this viewport comparison is done taking into account also the spherical geometry of the content. Due to its robustness and specificity for VR content, this tool has been used to analyse navigation patterns in some publicly available datasets such as [119]. The number of total clusters and users per cluster is a proxy of users similarity in exploring ODVs: the fewer the clusters, the more users are focused on a similar area of the video content highlighting a similar behaviour. This analysis has showed that video characterised by a dominant focus of attention (*i.e.*, moving objects) are explored in a similar way by users, except few exceptions due to camera motion [119]. For instance, video which have a vertical camera motion brings viewers not to be focused on a specific area but to be distributed on the landscape.

With the idea of developing an objective metric able to assess users similarity in a spatio-temporal domain, in Chapter 4 we propose a novel *affinity metric* to quantify the users similarities detected via the spherical clustering. This is evaluated as the normalised weighted average of cluster popularity (*i.e.*, how many users per cluster) and it equals 0 when users are not clustered together because highly dispersed in the navigation or 1 when users share a strong similarity in their trajectories. Equipped with this tool, we have noticed that the affinity between users is highly correlated to the selected VR device: HMD leads to higher *affinity metric* than other devices. A similar outcome has been also identified by authors in [134], which directly compare different prediction models for navigation trajectories collected by different devices such as HMD, PC and smartphone. Other than showing that exploration done via desktop is more static than the other devices, it has also

been verified a higher prediction accuracy for HMD users, especially in the near future. Following this research direction, a novel viewport clustering algorithm as a tool for behavioural analysis and video categorisation has been proposed recently in [121]. The main novelty is to consider multimodal clustering analysis, in which the spherical location of each viewer is augmented by other modalities as input such as head movement speed and the percentage of the sphere explored in a given time window. In this way, users in the same clusters are similar not only in terms of displayed viewport but also for the style of exploration.

Beyond clustering, we propose other tools to understand similarities and study user predictability. In Chapter 5, we carry out a behavioural analysis exploiting tools from information theory such as *actual entropy*, *transfer entropy* and *mutual information*, which quantify randomness and uncertainty in the users trajectories. For example, the actual entropy is low for users that experience "repetitive" behaviour (trajectories) over time, leading to highly predictable users. First, we used these tools to perform a intra-behaviour analysis, which is aimed at studying the behaviour of one single user across diverse contents. The analysis has shown that users can be profiled based on their interactivity: viewers tend to preserve similar navigation type (highly erratic or quite static) independently by the video content. Moreover, we also quantify a more discontinuity and randomness in navigation trajectories within ODVs lacking of specific FoA. As second line of investigation in Chapter 5, we present an inter-user behaviour analysis measuring how informative other users behaviour is for a current user. In other words, this study quantifies how much information about the predictability within a specific content can be extracted by the navigation of a given group of users. This analysis is similar to the previous one based on clusters and users similarity. However, the information theory tools capture a more meaningful behaviour and better quantify viewers similarity during their navigation than metrics based only on the spatial location of users, such us our proposed spherical clustering (Chapter 4).

3.4 User-centric ODV streaming

The tools and metrics discussed in the previous section enable a deeper understanding and prediction level of the users interaction, opening the gate to personalised and user-centric systems, Figure 3.2 (light blue and green boxes), where the different steps of the pipeline are tuned based on single user's behaviour. In the following, we review the main contributions that have been proposed toward user-centric systems, distinguishing the work based on the type of behavioural information: *i)* extracted from a single user, *single-user design*, *ii)* extracted from multiple users, *cross-user design*.

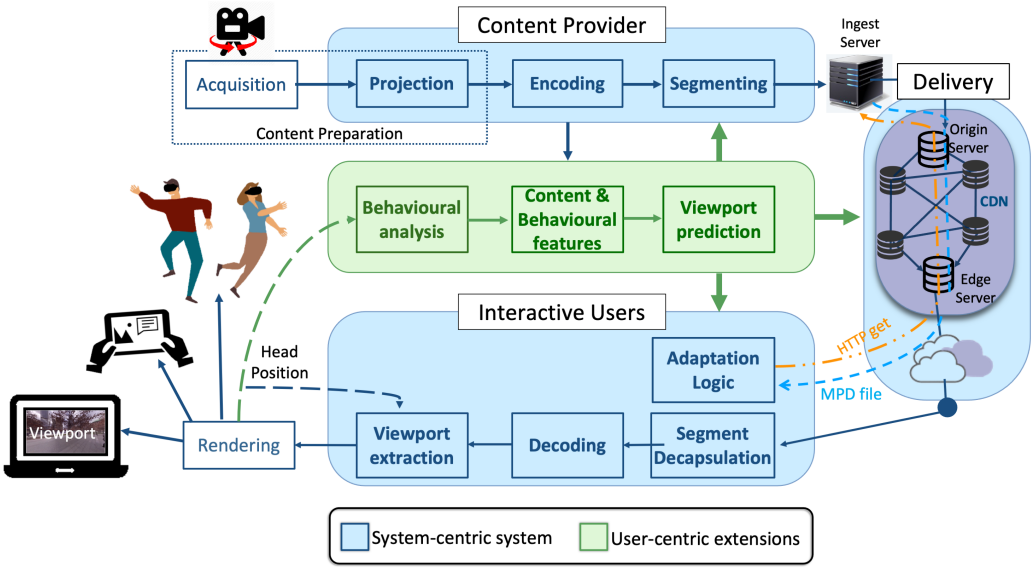


Figure 3.2: User-centric system pipeline.

3.4.1 Single-User design

A key step in user-centric systems is the prediction of the users head movement. The first and among the simplest technique (and most widely adopted) is based on the past and current trajectory of a single user, neglecting other viewers and video content information. Qian *et al.* [135] have experimentally shown how three simple logistic regression models, such as average, Linear Regression (LR) and Weighted LR (WLR) with a moving window of 1 second are able to successfully anticipate the user behaviour in the next short time window (*i.e.*, 0.5, 1, and 2 seconds). For the first time, authors are able to prove the potentiality of the prediction. Specifically, giving higher fetching priority to tiles that most likely will be displayed can reduce the bandwidth usage up to 80%. Multiple works have followed all adopting such simple predictive algorithms for user-centric adaptation logic, showing the gain in terms of bandwidth, re-buffering, and final quality experience by the user. Experimental validation has been proposed in [136], in which the user-centric adaptation logic has been tested on real-world 4G bandwidth. Results have shown that the proposed strategy maintains a good displayed quality (the same achieved when sending the entire panorama) but with a reduction of bandwidth overload up to 35%. Logistic regression based on historical data has been used also in [137], in which the adaptation logic for ODV streaming optimises the optimal representation per tile to request according to network bandwidth and predicted users' head movements. Their analysis emphasised the need of accurately predicting future viewport position for ODV streaming. A WLR algorithm is considered in a similar framework but in the case of scalable video coding [138]. Specifically, high-quality represen-

tations of tiles within the predicted viewport are prefetched shortly before being visualised to ensure high quality in the displayed content and at the same time to reduce storage costs at the server-side, which has been an open challenge especially for VR content that is highly data intensive.

Most of these works have shown the potentiality of user-centric systems, but suffer from poor prediction accuracy in the long-term (mainly due to the lack of other users and content information). At the same time, the behavioural studies highlighted in Section 3.3 have shown a strong consistency and similarity in the way in which users navigate ODVs, motivating cross-users designs described in the following subsection.

3.4.2 Cross-Users design

To breakthrough the limitation of single-user frameworks, a new research direction has been carried out aimed at exploiting behavioural information from *multiple* users to identify and predict the most popular trends in navigating ODVs and develop user-centric systems accordingly. In the following, we review these efforts describing first the work that are *content-agnostic* (only based on user cross-users information) and *content-based* (users information is augmented by content information).

Content-agnostic

At first, linear model and classical clustering have been widely used to infer single user behaviour from cross-user information [130, 133, 139, 140]. For example, with the main intent to improve the long-term users viewport prediction, Ban *et al.* [139] propose a viewport prediction approach based on K-Nearest-Neighbours (KNN) algorithm and aimed at combining both behavioural characteristics of the single-user and those extracted by benchmark viewers. Specifically, the algorithm is composed of two main steps: *i)* the user head position is predicted via LR model based on the historical movements of only the single user under investigation; *ii)* this prediction is then used to form the K nearest users set, as the users with the closest viewport centres previously collected. The K-NN set is used to compute the viewing probability per tiles. This is one of the proposed by Xie *et al.* [133], aimed at using cross-user information to identify main clusters of users and then predict new users mimicking the behaviour of the closest cluster. Per each detected group, the viewing probability of tiles is then computed and applied to support the viewport prediction during video playback of new viewers. Even if the prediction is more accurate in a time window of 3 seconds and longer, viewers are clustered based on Euclidean distance, neglecting the actual spherical geometry, resulting in not fully representative clusters. With the idea of exploiting the spherical geometry, other white-box models

have been proposed. For instance, Hu *et al.* in [141] use a graph-based approach to improve the accuracy of viewport prediction in a QoE-optimised ODV streaming system. Authors first predict tiles in the FoV by a tile-view graph learned from historical users navigation trajectories: a weighted graph is constructed in which each vertex corresponds to a tile while the weight is given by users behaviour. From the constructed graph, a tile view probability is finally evaluated and used to optimise the downloading bitrate per tile in a limited bandwidth system but maximising the users quality of experience. Authors deeply compare their proposed system with other algorithms (both navigation predictive and streaming) reaching 20% improvement in terms of users QoE.

Even if these first and simple approaches for predicting the user behaviour has very low computation complexity, they require to collect navigation trajectories from multiple viewers in advance for any ODVs making this strategy not always possible (*e.g.*, new video content). To overcome this issue, cross-users data analysis has opened the gate to deep learning frameworks aimed at inferring non linear interactivity models from a training dataset of collected trajectories based on supervised learning paradigms. These models have been augmented also by auxiliary losses [122] or by probabilistic models [81, 142] or state-of-the-art transformers [143] aimed at inferring the prediction error. For example, a Gaussian distribution based on previous immersive navigation experiences is used to model the distribution of short-term prediction error for new viewers in the system. This prediction approach is used to improve a viewport-dependent streaming system following a tile-based streaming approach in [142] and an improved coding technique (*i.e.*, pyramid projection) to adapt quality distribution based on users behaviour in [81]. Instead, Chao *et al.* [143] propose a viewport prediction transformer method for ODV, named 360° Viewport Prediction Transformer (VPT360), taking advantage of transformer architecture [144].

Content-aware

In the previous sections we have described the advances made toward user-centric systems, in the case in which user behavioural analysis was carried out by looking only at users data. However, the studies in Section 3.3 have shown that users attention is steered by the content as well. Therefore, in the following we review user-centric designs in which researchers have used both users trajectories and content features to infer user behaviour.

A very well known metric that maps content features into user attention is the *saliency map*, which estimates the eye fixation for a given panorama. Since a correlation between saliency map and user trajectory has been empirically proved

in [145], many efforts have been dedicated to study, infer, and exploit saliency in ODV streaming. Specifically, deep learning frameworks aimed at predicting users trajectories where augmented by using saliency maps as further input [115, 146, 147]. Different learning architectures and paradigms where considered in these studies: Reinforcement Learning (RL) based approach looking at the user's behaviour as sequential actions taken over time [115]; and recurrent learning approach exploiting the temporal correlation of users trajectories [146, 147]. Xu *et al.* [115] proposed an RL based workflow that first estimates the saliency map for each frame, and then, based on it together with historical data, predicts the viewport direction. This prediction is cast as a RL agent that aims at minimising the prediction loss (dissimilarity between the predicted and ground-truth trajectories). Its viewport prediction is however short-term being limited to the next frame only (*i.e.*, about 30ms prediction ahead). In the case of recurrent neural networks, Nguyen *et al.* [146] feed a Long Short-Term Memory (LSTM) network with both saliency (inferred by a CNN model) and historical head orientation from users. The learning framework was able to overcome main limitations such as central saliency bias and single object focus (*i.e.*, ODV users quickly scan through all objects in a single viewport). Interestingly, Rondón *et al.* [147] show that the historical data points (in terms of past trajectories) and content features may influence the future trajectories differently based on the prediction horizon. They observe that users trajectory is driven by the content more toward the end of the trajectory, which can be explained as at the initial phase of the trajectory users tend to have more erratic (and less content driven) behaviour. As a consequence, they propose a prediction model that prioritised user trajectory inertia first and visual content more at a later stage. Thus, different from other methods that only has time-dependence for the user positional features, they propose positional and saliency as two time series to feed an LSTM. These deep learning frameworks have strong potentiality but it is well known that they are data-hungry, with a tendency of poor training accuracy or lack of generalisation in the case of the limited datasets. Hence, works have been also presented in a parallel direction of "shallow" learning frameworks, such as the from Zhang *et al.* [148]. Authors designed their trajectory prediction as a sparse directed graph problem inferred by past users positions, saliency map data, and the biological human head model (which defines transition constraints on the graph given the physiological constraints such as impossible head movements).

Beyond saliency maps, other content features have been considered. In the case of dynamic scenes, for example, saliency might not be representative enough and *content motion* can be preferred as a feature. Such motion can be captured from either optical video flow [118, 149, 150] or from individually detected ob-

jects' movements [151, 152]. In [149], the optical flow as well as the saliency and the past trajectory are input to a LSTM-based prediction model. User-centric systems that exploit the proposed fixation prediction network achieve a reduction of both bandwidth consumption and initial buffering time. Deep learning frameworks have also been considered to extract content features and favour the temporal prediction of users viewing trajectories. Park *et al.* [153] implement a 3D-CNN to extract spatio-temporal features of ODVs and predict future viewing directions given both saliency historical users trajectories as inputs. The predicted trajectory is then exploited in a RL model that determines the downloading order and the downloading bitrate for tile-based streaming. Similarly, Xu Yanyu *et al.* [118] exploit the CNN architecture to extract optical flow and use the use users' gaze for the prediction. To predict users' gaze, authors created an eye-tracking dataset captured from dynamic scenes. The computed saliency maps and the content motion maps are in two spatial scales: at the entire panorama image; and at a sub-image centred at the current gaze point. Both saliency and motion maps feed a CNN for feature extraction, and then a LSTM predicts gaze direction using the current time and gaze point. Moreover, other works use individually detected objects' movements, mostly following the success of YOLO (You Only Look Once) [154] for objection recognition Chopra *et al.* [151] propose an online regression model based on trajectories of both users and the main objects, which are extracted online from the detection model. They claim that the user's head movement highly depends on objects trajectories. Their experiments highlight 34% model weighted given the object's trajectories.

3.5 Summary

Spherical contents have become widely spread with their first commercial application (*i.e.*, Google street view) out in 2007 and have attracted a growing attention in the multimedia community. As introduced also in the previous chapter, this novel multimedia format has revolutionised how users engage and interact with media content, going beyond the passive paradigm of traditional video technology, and offering higher degrees of presence and interaction. Thus, many new challenges have risen over the entire end-to-end communication chain due to the novel role of the user and the new geometry. For example, the spherical content needs to be efficiently delivered to the viewer taking into account also the aspect of user-content interaction and bandwidth limitation. In this context, this chapter has presented a summary of research advances in ODV adaptive streaming, mainly in terms of *user-centric* streaming solutions. Given the key role of the users, behavioural investigations on how viewers navigate within ODV have attracted a lot of interest, showing the benefit of understanding users behaviour, and enable personalised ODV

streaming solutions (*i.e.*, user-centric streaming system). Thus, this chapter has highlighted the main outcomes of these novel behavioural analysis, clearly distinguishing works in terms of a more *traditional data analysis* aimed at identifying general behavioural features, and *trajectory-based data analysis* focused on detective more specific key features (*e.g.*, similarities in interactivity).

The comprehensive survey presented in this chapter motivates and better identifies the research direction I have undertaken during my PhD, which is described in the following chapters.

Chapter 4

Spherical clustering of users navigating in VR content

In the previous chapter, we have shown the importance of understanding how users explore the VR content in order to optimise content creation [155] and distribution [82, 136, 149, 156], develop user-centric services [126, 157], user-based QoE assessment [158], and even for medical applications that use VR to study psychiatric disorders [128]. To enable behavioural analysis, in this chapter, we propose a novel tool (*i.e.*, spherical or graph-based clustering) aimed at identifying groups of users who behave in a similar way while interacting with immersive content. To emphasise the importance of this behavioural tool, we present also a case study of behavioural analysis across content and different VR devices.

4.1 Introduction

As described in Section 3.3, in the last few years, many studies have appeared collecting and analysing the navigation patterns of users watching VR content. Most studies build content-dependent *saliency maps* as main outcome of their analysis. The saliency map computes the most probable region of the sphere attended by the viewers, based on their head or eye movements [116, 156, 159–161]. Some works also provide additional quantitative analysis based on metrics, such as the average angular velocity, frequency of fixation, and mean exploration angles [109, 157]. However, none of these supply a quantitative metric to evaluate common patterns. On the contrary, clustering users based on their common navigation patterns could be a first direction to better understand users behaviour. In fact, performing a clustering of navigation trajectories can show how many groups of users consistently share a similar viewport over time. The evaluation of common portion (*i.e.*, overlapped viewport) of 360° content among users could be a key-metric to evaluate users behaviour. This information might be useful in order to improve the accuracy

and robustness of algorithms predicting users navigation paths. A proper clustering could also be useful to refine user-centric distribution strategies, where for instance different groups of users might be served with higher quality content in different portions of the sphere that will be more likely displayed by the viewers. In this context, the main goal of this chapter is to propose a novel clustering strategy able to detect meaningful clusters on the spherical domain. Specifically, we consider as meaningful cluster a set of users *attending the same portion of spherical content* (*i.e.*, a set of users with substantial overlap between viewports). The main motivation is that a significant common overlap needs to be guaranteed for clustering methods to be used for prediction purposes or for implementing accurate user-based delivery strategies.
To the best of our knowledge, studies identifying clusters for omnidirectional content delivery have appeared only recently [130, 133]. In [133], the viewing directions of each user, *i.e.*, viewport centers, are considered as points on the equirectangular planar. These are then clustered based on Euclidean distance, neglecting the actual spherical geometry. Conversely, in [130] each user navigation pattern is modelled as independent trajectories in roll, pitch, and yaw angles, and a spectral clustering is then applied. While it is efficient in discovering general trends of users navigation, this clustering methodology is not focused on identifying clusters that are consistent in terms of overlap between viewports displayed by different users. This means that users in the same cluster do not necessarily consume the same portion of content. Thus, the identified clusters are not necessarily meaningful in the perspective of studying behavioural navigation patterns.

At the same time, research interests have recently expanded toward psychological and emotional aspects related with VR applications. Since ODVs can be experienced by heterogeneous apparatus, such as smartphones, tablet and HMD, recent psychological investigations on users experiences suggest that viewers prefer different devices based on content category and their current location (*i.e.*, travelling or at home) [162]. Moreover, human perception is strongly dependent on the selected viewing platform [163]. From a technical perspective, the investigation of users behaviour in relation with selected content and device could be the key to optimise the system design of VR applications. However, currently, this is not possible since the behaviour of interactive users across devices is highly overlooked in the literature.
To overcome this issue, this chapter introduces a dataset of navigation trajectories of users watching 15 ODVs on different devices (HMD, tablet and laptop). As main novelty, we investigate different conditions of ODVs exploration based on the viewing device: traditional VR-based navigation enabled with HMD, touch-based navigation with tablet and mouse-based navigation with laptop. Based on this collected

dataset, we also propose a case study of behavioural data analysis across content *and* across viewing device. A first analysis is carried out with conventional metrics such as angular velocity and viewport center distribution, and it highlights the dependency of the users navigation from the displaying device. However, this first part of the analysis fails in detecting how much users interact in harmony among themselves; key information to understand users predictability. Therefore, we expand the dataset analysis including a novel metric aimed at evaluating the affinity among users – *i.e.*, the similarity among them in terms of viewport displayed overtime. Namely, we introduce the *User Affinity Index* metric. This allows us to move a step forward in the direction of better understanding how users interact with the VR technology, with a substantial impact on the efficiency of VR systems.

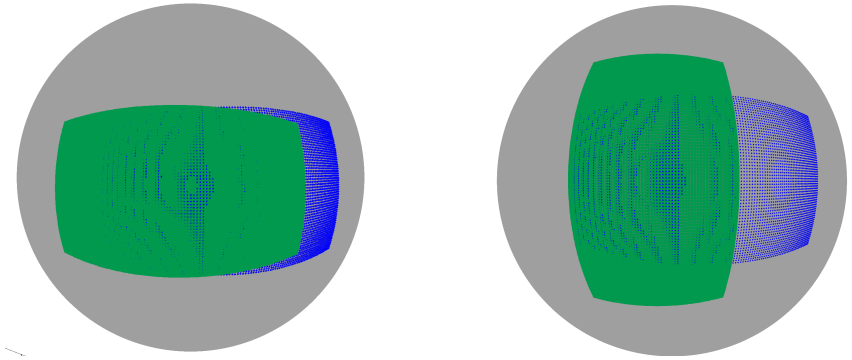
In conclusion, this chapter contributes to the overall open problem of behavioural analysis in 3-DoF VR system, with the following main contributions:

(a) A novel clustering algorithm that $i)$ considers the spherical geometry of the data, $ii)$ identifies clusters in which there is a consistent and significant geometric overlap between the portions of spherical surface corresponding to viewports attended by different users (by imposing that clusters are cliques), $iii)$ can be applied to a single frame or to a series of frames (trajectories).

(b) A new public dataset of 15 ODVs with associate navigation trajectories collected in task-free experiments using 3 different devices such as HMD, tablet and laptop.

(c) An exhaustive analysis of the aforementioned collected data, showing that users navigate differently based on the device, and introducing a novel affinity metric able to quantify user navigation similarities.

The remainder of this chapter is organised as follows. First we define a metric to quantify the geometric overlap between two viewports on the sphere (Section 4.2). Then, we use this metric to build a graph whose nodes are the centers of the viewports associated to different users. Two nodes are connected only if the two corresponding viewports have a significant overlap (Section 4.3). Finally, we propose a clustering method based on the Bron-Kerbosch (BK) algorithm [164] to identify clusters that are cliques, *i.e.*, sub-graphs of inter-connected nodes (Section 4.3). Results demonstrate the consistency of the proposed clustering method in identifying clusters where the overlap between the portions of the spherical surface corresponding to different viewports is higher than in state-of-the-art clustering (Section 4.4). As case study, we collect a novel dataset which is described and anal-



(a) 87% overlap **(b)** 58% overlap

Figure 4.1: Viewports (in green and blue) with $\pi/10$ centre distance. (a) viewports are aligned with an overlap of 87%, (b) one viewport is rotated by $\pi/2$ resulting an overlap of 58%.

ysed, respectively in Section 4.5 and 4.6. Finally, the chapter is summarised in Section 4.7.

4.2 Geodesic distance as proxy of viewport overlap

A key aspect of our clustering algorithm is to group users based on a metric that reliably reflects users similarities during the navigation. We argue that similarity in the navigation is captured by viewport overlap. In this section, we identify a metric that reliably reflects this overlap. More specifically, each user attends a portion of the spherical surface. This is the projection on the spherical surface of a plane tangent to the sphere (*i.e.*, *viewport*) in the point that identifies the user's viewing direction (*center of the viewport*)[1]. The overlap between the viewports attended by two users at an instant in time is a clear indicator of how similar users are with respect to their displayed viewports. For example, an overlap equal to the area of the viewport corresponds to two users attending exactly the same portion of visual content. The geometric overlap could be analytically computed, knowing the rotation associated to each user head's position (*i.e.*, roll, pitch, and yaw) and the horizontal and vertical fields of view that define the viewport. However, this is non trivial since it requires to evaluate closed-form expression of the viewport on the sphere. Here, we show that the *geodesic distance* between two viewport centres under specific settings acts as proxy of the viewport overlap. Thus, we propose the simple and straightforward solution of using this distance as a proxy for viewport overlap.

The geodesic distance is the length of the shortest arc connecting the viewport centers on the sphere. Such distance is an approximation of the actual area

[1]Without loss of generalisation, we consider a scenario in which the viewports of all users have the same horizontal and vertical field of view.

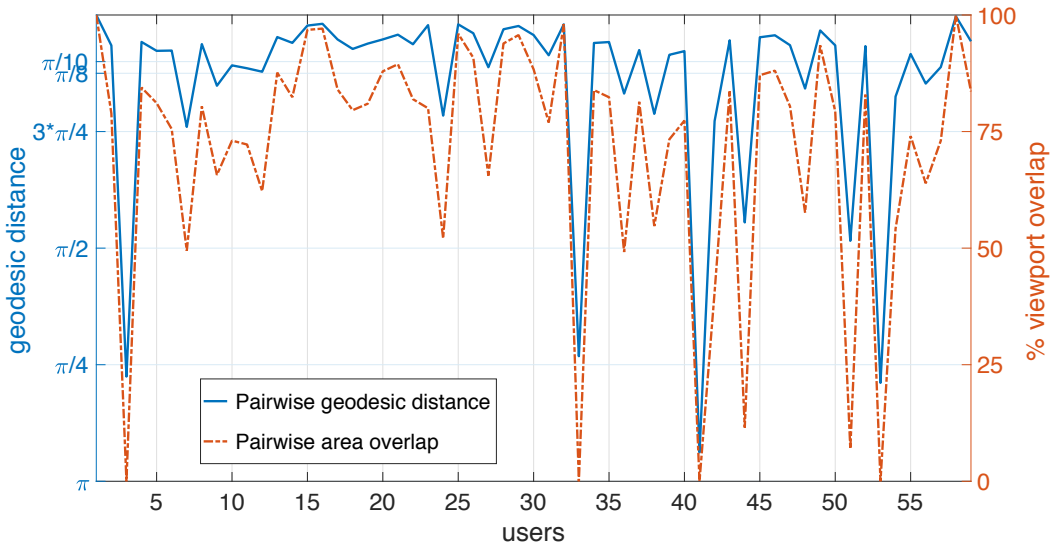


Figure 4.2: Comparison between pairwise geodesic distance and viewport overlap in one frame of Rollercoaster video.

overlap as it does not account for the 3-DoF of the user's head rotation. As a result, viewports whose centers have the same geodesic distance could correspond to a different viewport overlap due to the intrinsic approximation error (example in Figure 4.1). Nevertheless, the smaller the distance between viewport centers, the smaller the approximation error with geodesic distance. As an example, Figure 4.2 shows the pairwise geodesic distance (in blue) and the pairwise area overlap (in red) between the viewport attended by one user and those of 58 other users in a frame of a video sequence extracted from the public dataset proposed in [109]. The correlation between the two metrics is evident: if the overlap is high, the geodesic distance between the two viewport centres is low. Particularly, a viewport area overlap that is larger than 75% of the viewport area corresponds to a geodesic distance smaller than $3\pi/4$. We are therefore interested in identifying a threshold value below which the geodesic distance is a robust proxy of the viewports overlap.

To empirically define this threshold, we built the Receiver Operating Characteristic (ROC) curve as follows. We assume that two users are attending the same portion of content if their viewports overlap by at least O_{th} of the total viewport area. We then define a threshold value for the geodesic distance G_{th} such that users are attending the same content if their geodesic distance is below threshold. Anytime users are separated by a geodesic distance lower than G_{th} and the overlap of their viewport is less than O_{th}, we experience a false positive. Conversely, a true positive is experienced if users separated by a geodesic distance above the threshold but experience a viewport overlap equal or higher than O_{th}. Equipped with these definitions, we can compute the ROC by considering all the videos and uses na-

vigation patterns included in the dataset described in [109]. Figure 4.3 shows the curve obtained in our scenario with $O_{th} = 80\%$. On the x axis of the ROC curve there is the False Positive Rate (FPR), *i.e.*, the probability to have a wrong classification over the number of actual negative events. This rate should be as small as possible. On the contrary, the True Positive Rate (TPR) on the y axis represents the probability to correctly classify an event. The best value of geodesic distance is $\pi/10$ since it corresponds to a TPR value equal to 1, which in our application means a sure identification of viewports with an overlap of at least 80% based on the geodesic distance between their centers. Therefore, in the following we assume $G_{th} = \pi/10$ as a suitable threshold to reliably approximate the area overlap between two viewports by means of the geodesic distance between their centers.

4.3 Clique-Based Clustering Algorithm

We now describe the proposed clustering algorithm, aimed at identifying clusters of users having a common viewport overlap. We model the evolution of users viewports over a time-window T as a set of graphs $\{\mathcal{G}_t\}_{t=1}^{T}$. Each unweighted and undirected graph $\mathcal{G}_t = \{\mathcal{V}, \mathcal{E}_t, \mathrm{W}_t\}$ represents the set of users[2] navigating over time, where $\mathcal{V}$ and $\mathcal{E}_t$ denote the node and edge sets of $\mathcal{G}_t$. Each node in $\mathcal{V}$ corresponds to a user interacting with the 360° content at instant t. Each edge in $\mathcal{E}_t$ connects neighbouring nodes, where two nodes are neighbours if the geodesic distance between the viewport centers associated to the users represented by the nodes is lower than G_{th} , as defined in Section 4.2. The binary matrix W_t is the adjacency matrix of $\mathcal{G}_t$, with $w_t(i,j) = 1$ if users are neighbors. More formally:

$$w_t(i,j) = \begin{cases} 1, & \text{if } g(i,j) \leq G_{th} \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where $g(i,j)$ is the geodesic distance between the viewport centres of users i and j and G_{th} is thresholding value, introduced in Section 4.2.

Looking at the graphs over time $\{\mathcal{G}_t\}_{t=1}^{T}$, we are interested in clustering users based on their trajectories within a time window of duration T. In other words, we are interested in identifying users that have similar behaviour over time. With this goal in mind, we derive an affinity matrix A that will be the input to our clustering algorithm, similarly to other clusters of trajectories [131]. Each element of A is

[2]Without loss of generality, we assume that the set of users does not change over time. This covers also cases in which users devices are not synchronised in the acquisition time, as users positions are usually interpolated to create a synchronised dataset.

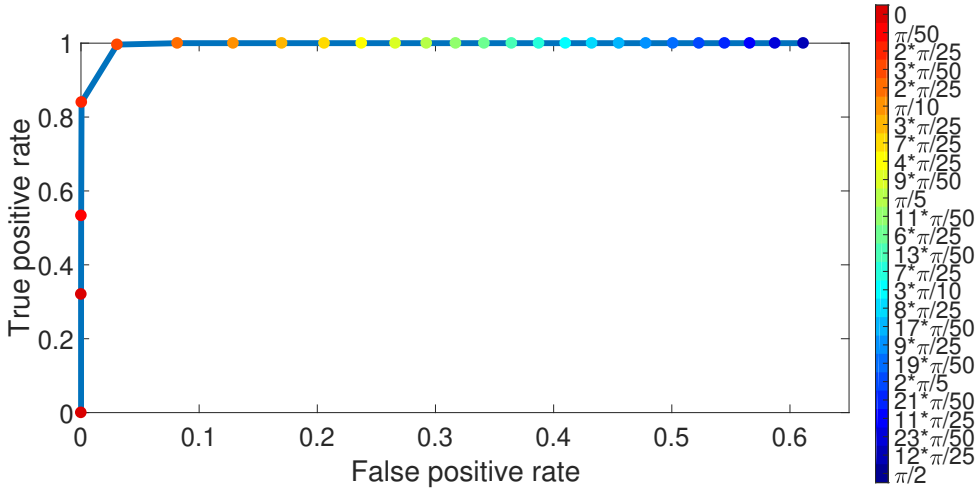


Figure 4.3: ROC curve to evaluate optimal G_{th} considering all video in analysed database and $O_{th} = 80\%$.

defined as following:

$$a(i,j) = \mathcal{I}_\tau \left(\sum_{t=1}^{T} w_t(i,j) \right) \tag{4.2}$$

where the function $\mathcal{I}_\tau(\cdot)$ is defined as $\mathcal{I}_\tau(x) = 1$ if $x \geq \tau$ and 0 otherwise. The matrix A can be associated to a trajectory-based graph where two nodes i and j are neighbours only if the corresponding viewports have a significant overlap in τ instants over T, *i.e.*, $a(i,j) = 1$. The more threshold τ approaches T, the more stringent the similarity condition.

As clusters, we want to identify group of users that are all neighbours (*i.e.*, $a(i,j) = 1$ for all pairs of users i and j belonging to the cluster). In graph theory, a set of nodes that are all connected to each other is called a *clique*. A clique perfectly matches with our definition of meaningful cluster: set of users all having significant pairwise viewport overlap, thus attending a common portion of video. Therefore, we propose a *clique-based clustering*. In particular, we consider the *Bron-Kerbosch (BK) algorithm* [164] to find all *maximal cliques* present in our graph (*i.e.*, the most populated sub-graphs forming cliques). While the BK algorithm identifies overlapping cliques (one user can belong to more than one clique), we are rather interested in identifying disjoint sets[3]. Hence, we build upon the BK algorithm and propose a clustering algorithm aimed at identifying non overlapping cliques, as depicted in Figure 4.4. We initialise the clustering method by evaluating the affinity matrix from Equation (4.2). Then, we perform the following steps (Algorithm 1):

[3]Clusters should be disjoint for most content-delivery applications. For example, if clusters are used for prediction, each user must belong only to one cluster.

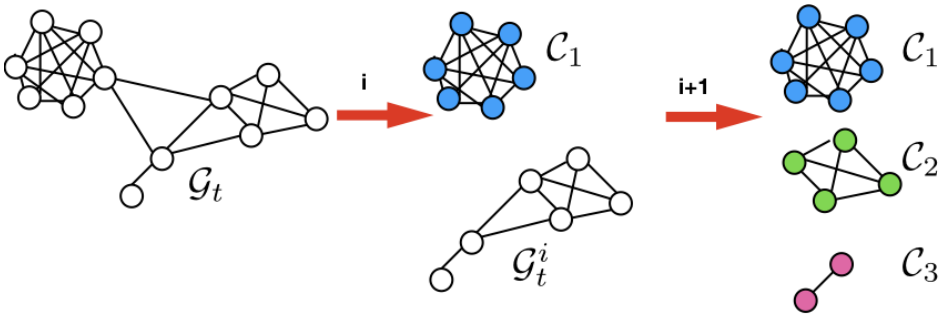


Figure 4.4: Graphical example of the proposed clique clustering.

1. Maximal cliques in the graph are detected by the BK algorithm.
2. Among the resulting cliques, only the most populated one (with the highest cardinality) is kept as a cluster.
3. A new affinity matrix is built, eliminating the entries corresponding to the elements of the cluster identified in Step 2.

These three steps are repeated until all nodes are assigned to clusters. It is worth mentioning that this iterative selection does not guarantee optimal clusters (*i.e.*, maximal joint overlap within the cluster). However, i) it imposes viewport overlap among users within a cluster, ii) it identifies highly populated clusters, which can be translated in reliable trajectories/behaviours shared among users.

4.4 Validation results

The proposed clustering algorithm is compared to state-of-the-art solutions, namely the *Louvain method* [165], the *K-means clustering* [166] and the clustering of VR trajectories proposed in [130] (labelled "SC"). We use the geodesic distance between viewport centers as distance metric in all algorithms. Moreover, in the *K*-

Algorithm 1: Clique-Based Clustering

Input: $\{\mathcal{G}_t\}_{t=1}^{T}$, D
Output: $K, \boldsymbol{Q} = [Q_1, ..., Q_K]$
Init: $i = 1$, $\mathrm{A}^{(1)} = \mathcal{I}_D(\sum_t W_t)$, $\boldsymbol{Q} = [\{\emptyset\}, \ldots, \{\emptyset\}]$
repeat
  $\boldsymbol{\mathcal{C}} = [\mathcal{C}_1, ..., \mathcal{C}_L] \leftarrow KB(\mathrm{A}^{(i)})$
  $l^\star = \arg\ \max_l |\mathcal{C}_l|$
  $Q_i = \mathcal{C}_{l^\star}$
  $\mathrm{A}^{(i+1)} = \mathrm{A}^{(i)}(\boldsymbol{\mathcal{C}} \setminus \mathcal{C}_{l^\star})$
  $i \leftarrow i + 1$
until $A^{(i)}$ *is not empty*;
$K = i - 1$

means clustering, the number of clusters K is imposed as the value achieved by the Louvain method (labelled "K-means 1"), as well as the K value obtained from our proposed clustering (labelled "K-means 2"). The proposed implementations have been made publicly available[4]. We test these algorithms on two video sequences 1-minute long (Rollercoaster and Timelapse), which have been watched by 59 users whose navigation paths are publicly available [109]. Rollercoaster has one main RoI (i.e., the rail) while in Timelapse, there are many fast moving objects (e.g., buildings, people) along the equator line.

Frame-based Clustering

First, we consider frame-based clustering, in which users are identified by their viewport centers in one given frame. Table 4.1 reports results in terms of number of clusters (K), mean viewport overlap computed within each cluster composed by at least three users, and viewport overlap within the most populated cluster, that we refer to as the main cluster. The viewport overlap within a cluster is the joint overlap across all users' viewports in the cluster. The mean overlap is computed by averaging the viewport overlap of all clusters with at least three users identified at a given frame. In Table 4.1, we also provide the percentage of users covered by clusters. The proposed algorithm always ensures the highest viewport overlap (on average always over 50%) with respect to the other methods. This is due to the implicit constraint that is imposed by the clique-based detection of the clusters. This constraint leads to the identification of clusters that are populated and yet meaningful (i.e., with large viewport overlap among users). For example, in Rollercoaster at frame $40s$, our algorithm identifies a main cluster grouping 35% of the population with a viewport overlap of 58.33%. This is much higher than the overlap of 24.20% (0%) in the main cluster identified by the Louvain (K-means) method. Beyond the accuracy, another important parameter is the percentage of the population that is covered by clusters with a significant number of users. These clusters are the most useful ones to allow predictions. For instance in Timelapse at frame $50s$, our method identifies a large number of clusters (29), which also includes single users clusters. Nevertheless, half of the population (51.70%) belongs to clusters with more than 3 users with high value of joint overlap (71.40%).

Trajectory-based clustering

Second, we test the proposed algorithm over a time-window of duration $T = 3s$ and similarity threshold $\tau = 1.8s$. In this case, we compare the proposed solution with SC algorithm [130]. The latter is applied in the following conditions: $i)$ to trajectories spanning the entire video as in [130], $ii)$ consecutive time windows of

[4]https://github.com/LASP-UCL/spherical-clustering-in-VR-content.

Table 4.1: Clustering analysis of users in three selected frames from Rollercoaster (a) and Timelapse (b). In brackets, the percentage of covered population.

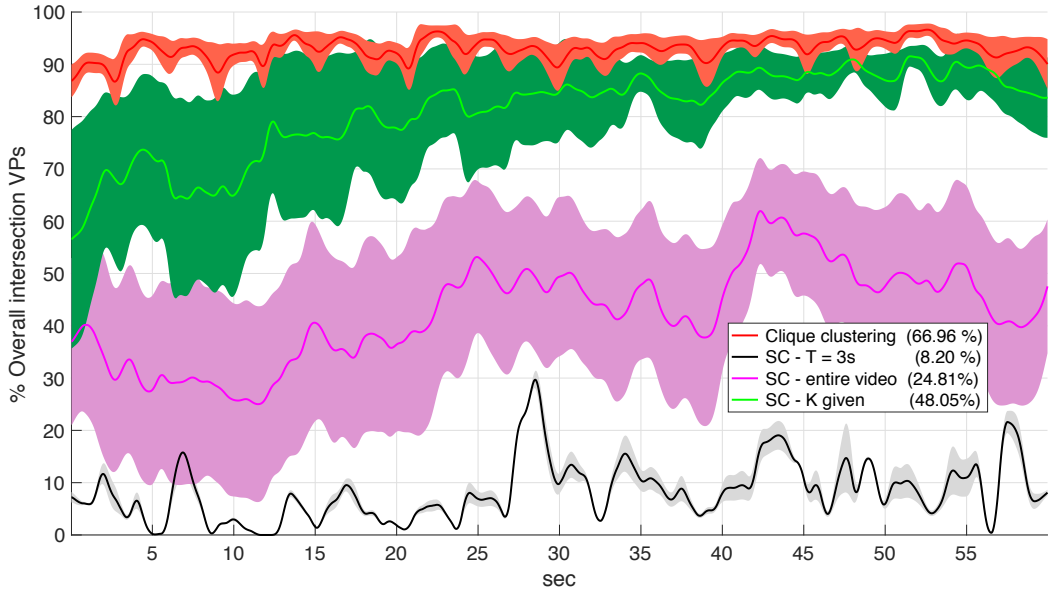
(a) Rollercoaster video

		ROLLERCOASTER			
		Louvain method	*Clique Clustering*	*K-Means 1*	*K-Means 2*
Fr. ~30s	K	10	15	10	15
	Mean Overlap Cl.(% user >3)	38.90 % (84.75 %)	**62.50 % (76.30 %)**	53.95 % (93.20 %)	48.10 % (94.90 %)
	Main Cl. overlap (% users)	26.70% (44.10%)	**58.60% (30.50%)**	48.30% (19%)	0% (20.70%)
Fr. ~40s	K	8	15	8	15
	Mean Overlap Cl.(% users >3)	35.60% (89.83%)	**65.75% (76.30%)**	44.38% (100%)	47.65% (84.75%)
	Main Cl. overlap (% users)	24.20% (45.80%)	**58.33% (35.60%)**	0% (30.50%)	0% (15.25%)
Fr. ~50s	K	8	12	8	12
	Mean Overlap Cl.(% users >3)	48.20% (89.80%)	**65.70% (86.45%)**	43.50% (98.30%)	55.30% (96.60%)
	Main Cl. overlap (% users)	46.40%(30.50 %)	**59.90% (57.70%)**	0% (22.40%)	0% (15.25%)

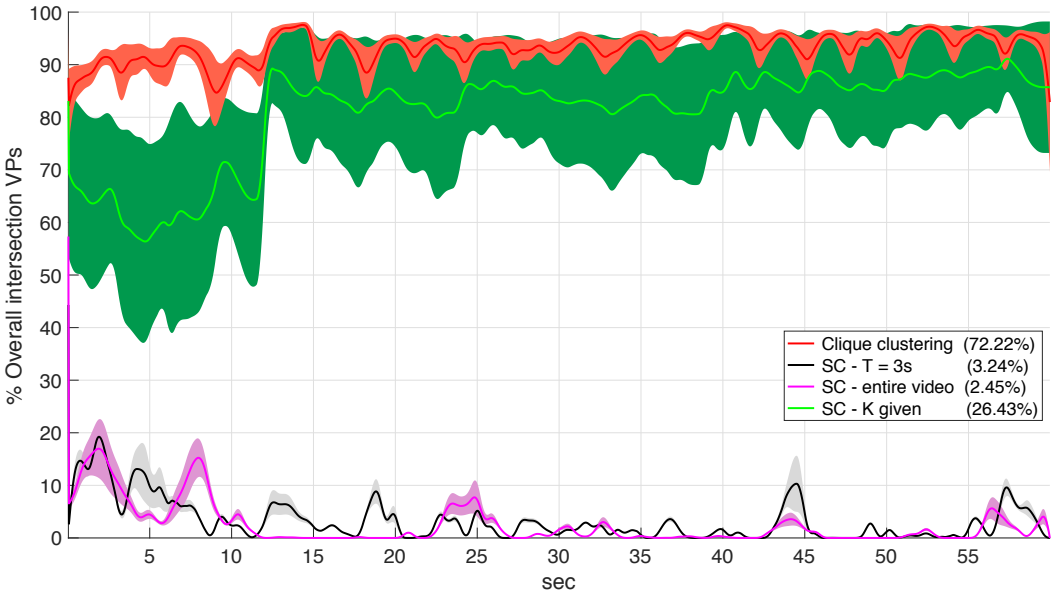
(b) Timelapse video

		TIMELAPSE			
		Louvain method	*Clique Clustering*	*K-Means 1*	*K-Means 2*
Fr. ~30s	K	13	24	13	24
	Mean Overlap Cl.(% user >3)	46 % (89.70%)	**72.35% (56.90%)**	45.90% (96.50 %)	51.50% (50%)
	Main Cl. overlap (% users)	32.90% (20.70%)	**69% (12.10%)**	15% (19%)	23.50% (13.80%)
Fr. ~40s	K	18	27	18	27
	Mean Overlap Cl.(% users >3)	47.65 % (75.90%)	**72.95 % (77.60%)**	60.27% (96.55%)	65.90% (84.50%)
	Main Cl. overlap (% users)	51.80% (20.70%)	**63.70% (17.24%)**	47.50% (20.70%)	33.60% (8.60%)
Fr. ~50s	K	18	29	18	29
	Mean Overlap Cl.(% users >3)	49.12 % (77.60%)	**71.40% (51.70%)**	48.36 % (87.90%)	55.90 % (55.17%)
	Main Cl. overlap (% users)	30.60 (22.40%)%	**70.80% (25.90%)**	37% (24.15%)	62.71% (17.24%)

duration T and $iii)$ imposing the same K obtained from our solution ("SC - K given"). Figure 4.5 shows results in terms of overlap among viewports clustered together in both Rollercoster (a) and Timelapse (b). In more details, all users are clustered over consecutive time-windows of T seconds each. Then, for each frame the viewport overlap among all users within one cluster is evaluated and averaged across clusters. The mean overlap (solid line) and the variance (shaded area) is finally depicted in Figure 4.5. Moreover, the mean value of joint overlap in clusters with more than three users across the entire video is shown in the legend. Our solution outperforms SC in terms of mean overlap but also in terms of variance. The latter shows the stability of our clustering method ensuring for each cluster a consistent overlap over time. Finally, the performance gain is significant also in terms of overlap in the most populated clusters (value provided in the legend).



(a) Rollercoaster video - T = 3 s.



(b) Timelapse video - T = 3 s.

Figure 4.5: Mean and variance of the joint overlap across clusters over time. In the legend, the mean value of joint viewport overlap of clusters with more than three users performed across the entire video.

Table 4.2: Description of the ODVs used for the subjective experiment. The dataset contains three content categories (documentary, action, and movie). Each content category has a training ODV and five test ODVs.

	Dataset ID	Name	Fps	YouTube Id	Selected Segment
Documentary	**Test**	*WildDolphins*	25	BbT_e8lWWdo	00:44 – 01:04
	01	*BabyPandas*	24	0XrH2WO1Mzs	02:05 – 02:25
	02	*Symphony*	30	LZlNCAGWtwE	01:10 – 01:30
	03	*Ocean Shark*	24	aQd41nbQM-U	00:40 – 01:00
	04	*Dancing*	24	raCda6VRrE8	00:00 – 00:20
	05	*Survivorman*	30	OLQzLOd7Xpk	00:30 – 00-50
Action	**Test**	*LaRonde*	25	r-qmDDi8S5I	00:10 – 00:30
	06	*FighterJet*	25	NdZ02-Qenso	00:00 – 00:20
	07	*HollywoodRockit*	25	Js_Jv5EzOv0	00:10 – 00:30
	08	*GetBarreled*	30	7gjR60TSn8Q	01:22 – 01:42
	09	*KITZ*	30	KS9S1Hgx2co	00:00 – 00:20
	10	*Knockout*	30	0x16ngo8xfY	01:22 – 01:42
Movie	**Test**	*Starwars*	25	SeDOoLwQQGo	02:23 – 02:43
	11	*Back2theMoon*	30	BEePFpC9qG8	00:11 – 00:31
	12	*Help*	30	G-XZhKqQAHU	01:20 – 01:40
	13	*Nick*	24	Au5ro1NOnh	03:25 – 03:45
	14	*Invasion*	25	QolJrTXr7PA	00:44 – 01:04
	15	*InvisibleMan*	25	I_FUpUi2LBk	01:55 – 02:15

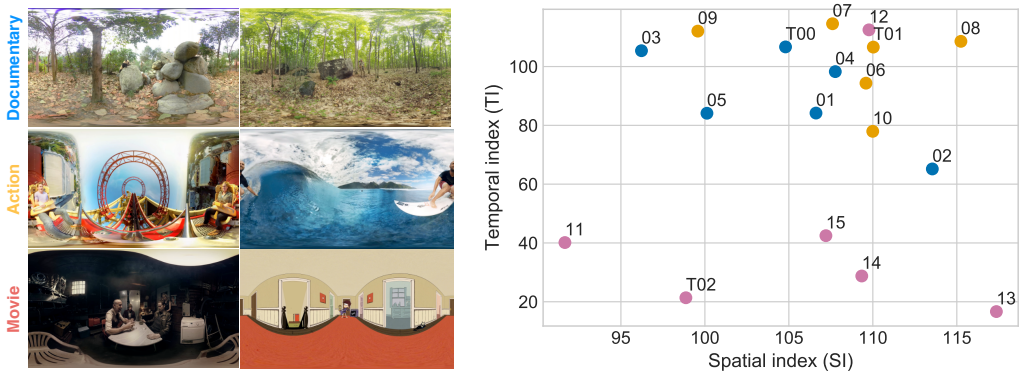
4.5 Collection of users navigation trajectories

As case study, we are primarily interested in understanding users navigation across space and time when interacting with different ODVs and the impact that different devices might have on the actual interaction. With this aim in mind, we collected a dataset with head-trajectories across different viewing platforms. In particular, we conducted subjective experiments across two universities, namely, Trinity College Dublin (TCD) and University College London (UCL). In this section, we describe the technical details of the experiments. The collected navigation trajectories and the used tools are shared in a public repository[5] under the MIT open source license.

4.5.1 Material

To ensure diversity in terms of content, we selected 18 ODVs with diverse content characteristics and representative of three video categories: *Documentary*, *Action*, and *Movie*. These categories are diverse enough to maximise the number of subjective experiments to carry out, and yet they span various content characteristics. Moreover, these categories are widely used in the classification of ODV content types. Figure 4.6 (a) depicts a snapshot of two randomly sampled ODV for each category. Specifically, we selected videos to span a wide range of content char-

[5] https://v-sense.scss.tcd.ie/research/3dof/vr_user_behaviour_system_design/



(a) Sample thumbnail frames of the ODVs (two per category) used in the experiments.

(b) SI and TI [167] of ODVs used in subjective experiments. Each category is visualised using different color: Blue: documentary; yellow: action, pink: movie.

Figure 4.6: Sample frames and statistics for the used ODVs in this work.

acteristics, such as spatial and temporal complexities. Fig 4.6 (b) visually reflects the diversity that each video exhibits in terms of spatial and temporal information measures [167], SI and TI, respectively.

Each ODV was downloaded from YouTube in the ERP format at the maximum available bitrate and resolution, which is 2560×1440. These ODVs were selected by a consideration of downloading ODVs with high quality. Then, a visual segment of 20 *sec.* duration was extracted from each video, and the audio signal was discarded from each ODV. Our work focuses only visual (texture) part of ODV by ignoring audio in every step of the delivery pipeline. In particular, we are interested at studying the effect of visual content on the trajectories, which has been the case in many other related works (*e.g.*, [109, 113, 116]).

Each 20 *sec.* segment was selected in a pilot test with two experts. The experts selected the 20 *sec.*, making sure that the selection exhibits its content category and contains at least one salient object. This duration was chosen as it is the most commonly used in visual attention studies [116]; specifically, it is a meaningful duration for the visual attention experiments as it is long enough for users to engage with the content, and yet short enough to maximise the number of experiments to carry out. Finally, an ODV from each category (out of the 18 ODVs) was used as training content for participants to familiarise with the setup of each device. Table 4.2 summarises characteristics of ODVs used in this work, where *Test* denotes the training content, one for each category.

4.5.2 Apparatus

We modified the JavaScript-based test-bed developed in [113] allowing users to display ODVs on three different devices, namely, HMD, laptop, and tablet, while

recording their navigation (*i.e.*, viewport center) trajectories for the whole duration of the experiment. The developed test-bed records participants' viewport positions with the current time-stamp and ODV name. Here, a given set of ODVs is first loaded using the playlist file, and a given video is played while the recorded data is transmitted to the server with the refresh rate of the device's graphics card. At the server side, the HTTP server was implemented using the Apache web server with the MySQL database, where the device-related (*e.g.*, HMD, laptop, and tablet), sensor-related (*e.g.*, viewing direction), and user-related (*e.g.*, user ID, age, and gender) data are stored on the database.

We conducted ODV subjective experiments with VR-based navigation enabled with HMD, touch-based navigation with tablet and mouse-based navigation with laptop. As HMD, we used the Oculus Rift consumer version that allows rendering of scene with a nearly 110 FoV at 90 Hz refresh rate. Each ODV is displayed in the HMD using the Firefox Nightly (*ver.* 67.0a1) Web browser. Finally, Alienware 15 Gaming Laptop and Apple iPad Pro 10.5 tablet were used. In both devices, we utilised Google Chrome (*ver.* 71.0.3578.98) as a web browser to play ODVs. We considered two different web browsers due to hardware and video codec compatibility issues at the time of subjective experiments.

4.5.3 Participants

In all, 94 participants (65 males and 29 females - about 30% women) took part in our subjective experiments. Participants were aged between 21 to 54, with an average of 31 years. Nine of the participants (about 10%) were familiar with ODV, and the others were naive viewers. Furthermore, 43 participants wore glasses during the experiment, and all of the viewers were screened and reported normal or corrected-to-normal visual acuity. Each participant watched a total of 18 ODVs (5 test plus 1 training ODVs per device).

4.5.4 Viewing procedure

To ensure diversity in participants (*e.g.*, ODV familiarity) and maximise the number of navigation trajectories, we performed the data collection campaign using the same apparatus at TCD and UCL. Each subjective test was performed as *task-free* viewing sessions in laboratory condition, where each participant was asked to naturally look at each ODV. The task-free viewing is the most common procedure for analysing visual attention [113, 168]. Participants were seated in a swivel chair and allowed to turn freely. During experiments, subjects were alone in the room to avoid any influences given by the presence of instructor.

The subjective test was divided into three phases, where in each phase a viewing session with a different device was conducted. The order of the devices was in a

random order. To get familiar with the new device, a training video was displayed at the beginning of each phase (one for each device). Then, after the test session, 5 test ODVs were played in a random order while the individual navigation trajectories were recorded using the implemented test-bed. To avoid motion sickness and eye fatigue, we inserted a 5 *sec.* rest period with a grey screen between two successive ODVs. Also, before playing each video, we reset the sensor to return to the centre of the ERP. In total, each viewing session lasted 2 *min.* and 25 *sec.*. We also set a 3 min break between each viewing session.

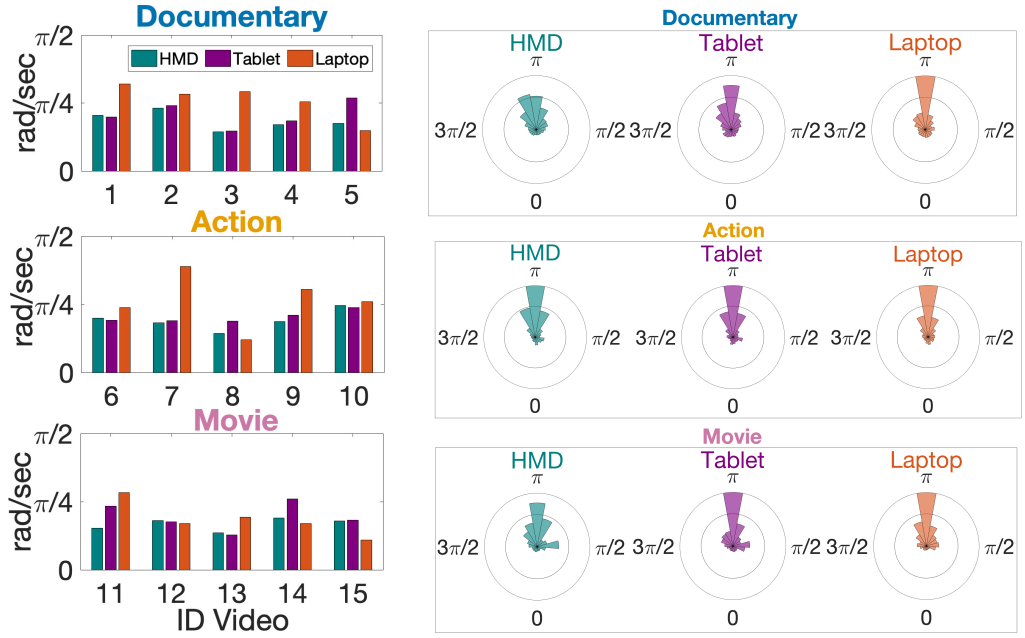
To ensure both the balance among the collected dataset (*i.e.*, balanced amount of viewport trajectories per device per video) and that each user watches each video only once, a set of playlists was prepared. Each playlist included a training and 5 test ODVs per device, and in total there were three different playlists for the three different phases of the test (*e.g.*, three different devices). These playlists were randomly selected for each user at the beginning of the subjective experiment. It is worth noting that the avoidance of repetition of the same video within the same playlist avoids the memory bias effect, that could affect the navigation trajectories [169]. Therefore, during one experiment, a user switches devices every 5 ODVs. In total, subjects watches 15 different ODVs.

4.5.5 Post-processing

In order to analyse the collected navigation trajectories, all recorded data was re-sampled based on the frame rate of the corresponding video. In this way, a fair comparison is allowed having a single value per user in each frame. Since roll movements are permitted only with HMD, our following investigations are based only on viewport's movements in longitude and latitude coordinates. Previous works [127] showed that most of the users' movements happen mainly along with these directions and the roll movements are at minimum. Therefore, this choice will not compromise the validity of the analysis presented in the following.

4.6 User behaviour analysis

We now present an analysis of the collected navigation trajectories across video content and devices. Specifically, we propose two lines of analysis: one more traditional aimed to show similar features of navigation among users, and the second focused on quantifying behavioural similarity among users. Over the entire section, we will also underline the key insights that we observe when users navigate in different video categories and with different devices.



(a) Angular velocity **(b)** Viewport center distribution

Figure 4.7: Traditional analysis of users' behaviour across devices and video categories. **(a)** Angular velocity per video and device - Video ID refers to Table 4.2. **(b)** Viewport center distribution on the longitude direction per video category and device.

4.6.1 A Conventional Data Analysis

We take the liberty to denote this first analysis of the collected trajectories as "conventional" data analysis since we adopt well know metrics such as angular velocity and spatial distribution of viewport center. Here, the key novelty is to investigate the users behaviour across categories *and* devices, leading to the following observations (supported in the remaining of the subsection):

- **Observation 1**: Users tend to be more dynamic with laptop compared to other devices.
- **Observation 2**: In contents characterised by a dominant focus of attention, the level of interactivity is negligibly affected by the displaying device (highlighting the dominance of the focus of attention).
- **Observation 3**: On the contrary, in contents with no focus of attention, users have highly exploratory trajectories and, a strong dynamic with HMD.

In Figure 4.7 (a), we analyse the users behaviour via the mean angular velocity per user for different devices and video categories. This analysis reveals the dynamicity of users navigation, measuring how fast each participant moves their head

inside a given ODV. It is worth noting that the angular velocity is typically lower using HMD rather than other devices; on the contrary, users experience the highest mean angular speed when displaying ODV on laptop. This can be motivated by the physical constraints imposed by HMDs (*i.e.*, limited head movements), but also by a deeper feeling of immersion experienced with HMD compared to the laptop. Implicitly, a drop of attention or immersion sensation leads to a more scattered navigation paths. Authors in [170] show how film editing and style influence user gaze movements during the vision of standard 2D movies. Therefore, comparing different video categories in our analysis, we can observe a slower angular velocity for users displaying *Movie* videos. This confirms that film maker manages to drive users' visual attention toward the main subject of interest also in ODVs. On the contrary, *Documentary* videos usually lack of a main focus of attention; hence, viewers tend to explore more the content.

Beyond the velocity of participants movements, we are also interested in detecting the areas of saliency, *i.e.*, the most interesting areas in which users look at. Figure 4.7 (b) shows the distribution of viewport centers in the longitudinal direction for all video categories across devices. Each slice, spanning a $\pi/10$ angle, represents the popularity of each direction across the entire video. The extension of each slice is proportional to the times in which - on average - users centred their viewports in the longitude direction identified by the slice. In particular, a single triangle predominant over others reflects that most of the users tend to center their displayed viewport in the same region of the ODV, identifying a clear focus of attention. From the figure, it is evident that the privileged area in terms of longitude is not really affected by video category and device. Viewers indeed tend to spend most of the time in a restricted portion of the central area around π in all different settings and video of the test. As expected, this is evident in *Action* and *Movie* categories, while less present in the *Documentary* contents, that usually have a less dominant focus of attention. In this latter case, the interaction is device-dependant, with a more spread distribution of viewport's centers with HMD when compared to laptop and devices.

4.6.2 Looking for Users Similarities

The metrics applied for a conventional data analysis reveal general and useful features of users behaviour. However, they do not necessary provide an answer to one simple and yet crucial question: *"Can we predict users behaviour?"*. Without pretending to fully answer to this question with the following data analysis, we truly believe that a key information to grasp is *"Do users behave similarly?"*. This is the key as users with poor similarity in the navigation are highly challenging to predict.

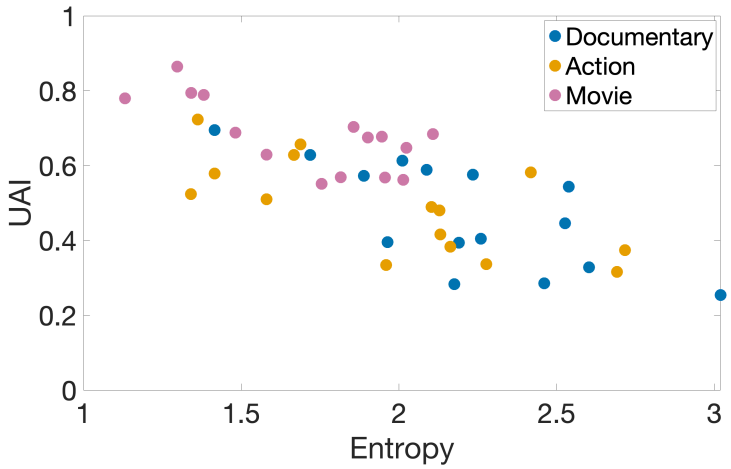


Figure 4.8: Comparison of UAI with entropy of saliency maps per each video of the entire dataset.

This motivates the following analysis, aimed at identifying behaviour similarities among users, across video content and/or devices; hence, the importance of developing metrics able to capture this information. Specifically, we analyse our dataset with the clique-based clustering algorithm presented in Section 4.3, which is able to identify users clusters based on their consistency in the navigation. In practice, the algorithm detects and puts together users that consistently display similar viewports over time while consuming the ODV content. Also, this is done by taking into account the spherical geometry of the ODVs. We therefore introduce a novel metric (based on the clique-based clustering algorithm) to better reflect similarity among users navigation trajectories within the same given ODV. We define this metric as *User Affinity Index (UAI)*, given as follows:

$$UAI = \frac{\sum_{i=1}^{C} x_i \cdot w_i}{\sum_{i=1}^{C} w_i} \tag{4.3}$$

where C is the number of clusters detected in a frame by the clique-clustering[6], x_i is the % of users (*i.e.*, out of the whole population/users sampled) in cluster i and w_i is the number of users in cluster i. In other words, the UAI represents the weighted average of cluster popularity (*i.e.*, how many users per cluster). The UAI approaches 1 when a small number of clusters with a large number of users per cluster are detected. This shows high affinity among users (*i.e.*, users share strong similarity in how they navigate the content). On the contrary, UAI tends towards 0 when participants experience highly scattered navigation patterns, and they cannot be clustered together.

To check the validity of our proposed metric, we evaluate also a well-known

[6]The clique-based clustering is applied with a geodesic distance threshold equal to $\pi/8$.

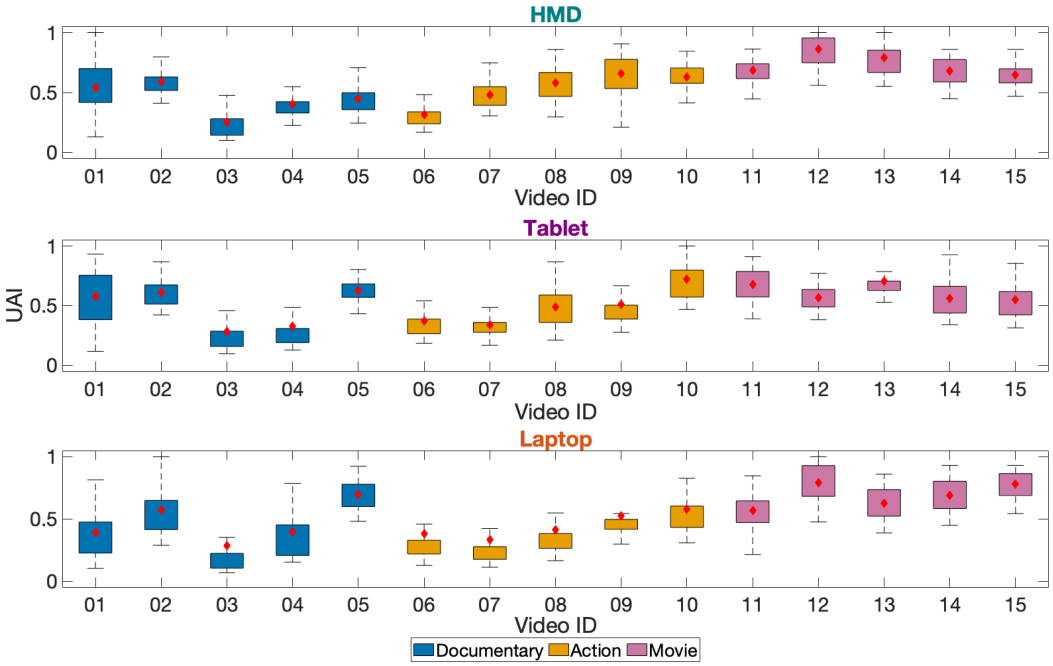
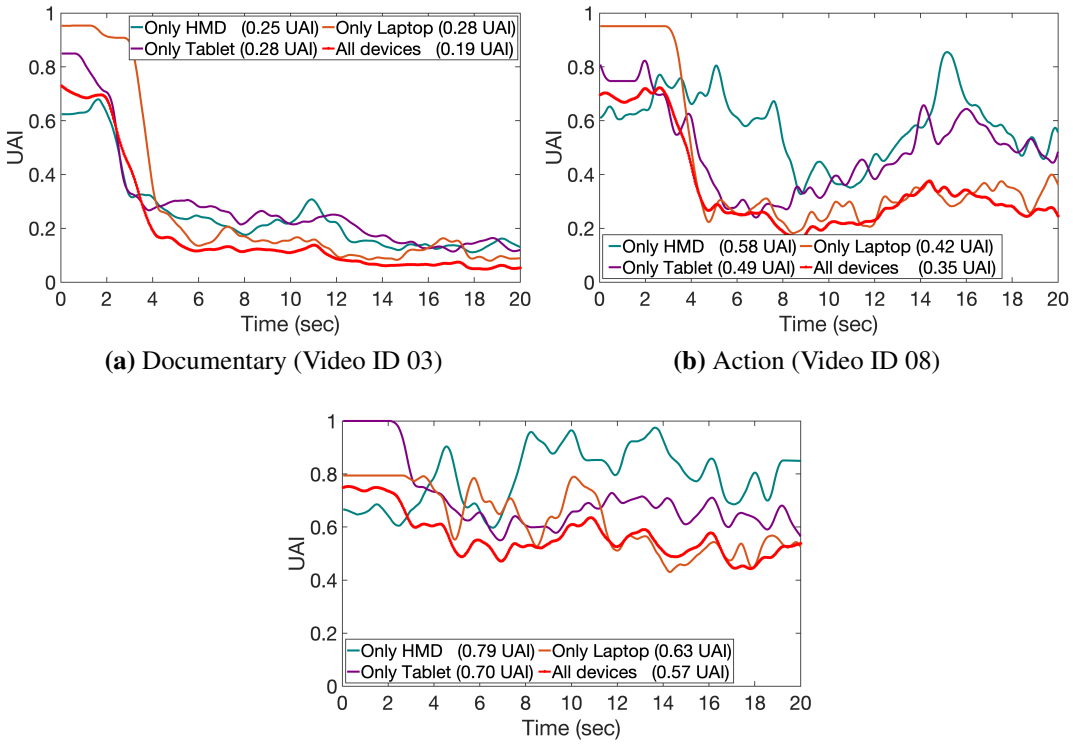


Figure 4.9: Boxplots per viewing device of Users' Affinity Index (UAI) for each video in the dataset. The lower and upper side of the rectangular represents 25% and 75% percentile, respectively. While diamond is the mean value of UAI per the entire video.

metric (*i.e.*, the entropy of the saliency map [158]) per each video of the entire dataset. This metric is typically used to evaluate model of visual attention, and gives a qualitative idea about the dispersion of users movements over time. In particular, low value of entropy stands for users focused all on a restricted area (*i.e.*, focused content - high correlation among users); while high value means more exploratory movements (*i.e.*, exploratory content - low correlation among users). Moreover, authors in [158] have applied this metric to omnidirectional images providing its validity also for this kind of content. Figure 4.8 shows the correlation between UAI and the entropy of saliency map per each video of the dataset averaged per devices. We can therefore notice a strong correlation between this traditional metric and our UAI. As expected, video characterized by low entropy have also high values of UAI meaning that users move similarly within the content; on the contrary, videos where users navigate more randomly, present high value of entropy.

Figure 4.9 shows the range and mean value (*i.e.*, box and red diamond, respectively) of UAI distributions obtained for each ODV of the entire database. Different behaviour can be identified based on the device and the category of video. For instance, the affinity for *Documentary* videos is lower than the one experienced with ODVs from the *Movie* category. We can also generalise that the navigation affinity within *Documentary* videos is not really influenced by the viewing device.



(a) Documentary (Video ID 03)

(b) Action (Video ID 08)

(c) Movie (Video ID 13)

Figure 4.10: UAI over time for three different videos (one per category) and for all devices. The mean value over time is reported on bracket in the legend for each analysed clustering condition.

On the contrary, HMD enable users to enjoy very similar experiences within ODVs, mainly for *Movie* and *Action* sequences. For example, users that display *Action* video with HMD have an UAI higher than 0.5 (except for Video ID 06). These findings are strongly evident in Figure 4.10 that shows the UAI over time for three selected videos, one per category (*i.e.*, ID 03, 08 and 13). After an initial phase where most of the users are focused on the same area, people start exploring the scene and behave differently based on the content (or video category). Specifically, in *Documentary* sequence (ID 03) users have a very low affinity, while they navigate in a much more compact way in *Movie* video (ID 13) leading to higher UAI for all devices (Figure 4.10 (a)). Moreover, HMD leads to more similar navigation paths compared to the laptop, see Figure 4.10 (b) and Figure 4.10 (c). Finally as a further comparison, we also apply the clique-clustering to all the recorded data without distinguishing them based on viewing device. We then evaluate the corresponding UAI (labelled as "All devices" in Figure 4.10) and notice that the affinity drops drastically, with respect to the case in which the clusters were formed per device. The "All devices" curve seems to be a worst-case scenario, showing that the users navigation has a strong affinity when looking at data from the same viewing device

but this affinity drops when analysing data for the same content but across devices.

In summary, from this second analysis we can conclude the following:

- **Observation 4**: In content with no main focus of attention, users experience a low affinity, which is interestingly not perturbed by the viewing device.
- **Observation 5**: Users tend to explore content characterised by a dominant focus of attention in a very similar way.
- **Observation 6**: In content with a main focus of attention, the user affinity is strongly related to the selected viewing device. In particular, the HMD leads to quite similar navigation among users.

These outcomes highlight the importance of studying navigation trajectories in VR systems per viewing platform. Specifically, similar users behaviours (*i.e.*, high value of affinity) identify predictable patterns that can be used to optimise user-centric streaming systems.

4.7 Chapter Summary

In this chapter, we introduced our graph-based clustering strategy able to detect meaningful clusters, *i.e.*, group of users consuming the same portion of a virtual reality spherical content. Our aim was not only to understand if users are looking in similar directions at the same instant. We were rather interested in understanding if users are displaying a significantly similar portion of the video within one frame or multiple frames. Firstly, the key challenge has been to define a proper value of geodesic distance that ensures similarity among users in terms of overlap. Then, we have proposed to use an important proprieties of graph theory such as *cliques* to identify clusters in a customise graph based on the previous distance metric. Results carried out on real VR user navigation patterns show that the proposed method identifies clusters with higher joint overlap than other state-of-the-art clustering methods. At the time of publication, the associated code has been made publicly available for future comparisons and to encourage the community to use our tool. As a case study, we also applied our graph-based clustering for a behavioural analysis aimed at exploring the way in which people navigate with omnidirectional video. To reach this goal, we conducted a subjective test across two different European universities (*i.e.*, UCL and TCD), collecting navigation trajectories with three different VR devices (HMD, laptop, and tablet). The collected data have been exhaustively analysed, showing key differences of users behaviour across device and content categories. For instance, users watching content from the *Movie* category

or displaying ODV with HMD will experience a more similar interaction between each other with respect to the case of other devices or other contents. The key novelty of this chapter was also the proposed *user-affinity metric* (UAI) aimed at evaluating the affinity among users.

This chapter allowed us to move a step forward in the direction of better understanding how users interact with 3-DoF VR technology. An alternative approach of behavioural analysis will be presented in the following chapter of this thesis.

Chapter 5

An Information-Theoretic analysis of immersive users

This chapter aims at advancing the understanding of 3-DoF VR users proposing a novel methodology and highlighting the importance of looking at users trajectories instead of more qualitative measures of user's interactions. By studying VR trajectories across different contents and through information-theoretic tools, we aim at characterising navigation patterns both for each single viewer (profiling individually viewers - *intra-user analysis*) and for a multitude of viewers (identifying common patterns among viewers - *inter-user analysis*). For each of these proposed behavioural analyses, we describe the applied metrics and key observations that can be extrapolated.

5.1 Introduction

Despite a growing attention on anticipating viewer's movements during an immersive experience [106], an efficient prediction tool is still an open research. As introduced in the previous chapter, one major limitation is the lack of understanding of user's behaviour in a VR experience, which could be crucial for an efficient prediction. For example, it is still not clear if the content has a dominant impact on user's navigation patterns; or if some users are more predictable than others. So far, the way in which users explore the VR content has been characterised in terms of angular velocity, frequency of fixation, and mean exploration angles [109, 157]. A more recent visual (and qualitative) tool used to study user's behaviour in VR is the heatmap, which identifies areas of the content mostly attended by viewers within a time interval [157, 171]. While these metrics and tool provide a general understanding of user's behaviour, they all fail in identifying similarity among viewers over time. For example, given a scene characterised by two FoAs, we can identify two types of behaviour: users that move continuously back and forward from the

two FoAs; others that display for a consistent interval the first FoA and afterwards move on the second one. On average, both types would spend the same amount of time displaying the two FoAs, leading to the same heatmap, despite their different navigation paths. Another possible metric to consider is the angular velocity, that could quantify the head motion speed, neglecting however the qualitative movements of users. In summary, a proper quantitative metric for user's behaviour study in VR is still missing.

The analysis of trajectories in a 3D space is a common problem widely investigated across many disciplines. For example, human mobility is a multidisciplinary field of social science, neuroscience and transportation, that refers to movements of people in a spatio-temporal dimension, such as the daily life on earth's surface [172]. A common trend has been to adopt Information-Theoretic (IT) metrics to statistically characterise the uncertainty of human mobility patterns [173]. *Information theory* is indeed an important tool born for communication systems, which has been used in different domains to detect hidden interactions in complex systems [174].

In this chapter, we attempt to use tools from information theory to identify key behavioural aspects of users during an immersive experience. We are interested in *quantifying* similarities not only among different users but also for the same viewer across contents, leading to a two-fold investigation: an *intra-user behaviour analysis*, and an *inter-user behaviour analysis*. To the best of our knowledge this is the first work using IT metrics for analysing trajectories in VR context. Until now, entropy has been applied only to heatmap [158] and not to user's trajectory as presented in this work. The intra-user behaviour analysis is aimed at understanding the level of interactivity of each single user across different contents. This shows that users tend to have an intrinsic identity in their interaction with the content suggesting the possibility to have user's profiling. On the other hand, the inter-user analysis considers navigation across an entire group of viewers to asses if user's behaviour can help in the prediction of other viewer's behaviour. We strongly believe these investigations can bring key-information in the understanding of any hidden patterns of immersive user's navigation. Our outcomes can be eventually exploited in algorithms to accurately predict where users most likely look at in the near future during an immersive experience. The remaining part of this chapter is organised as follows: an overview of the proposed VR analysis framework is provided in Section 5.2. Section 5.3 describes IT metrics considered in this work. A deep user's analysis is presented in Section 5.4, highlighting both similarities in the history path of a single user and across an entire set of viewers; the main outcomes of this behavioural study are finally summarised in Section 5.5.

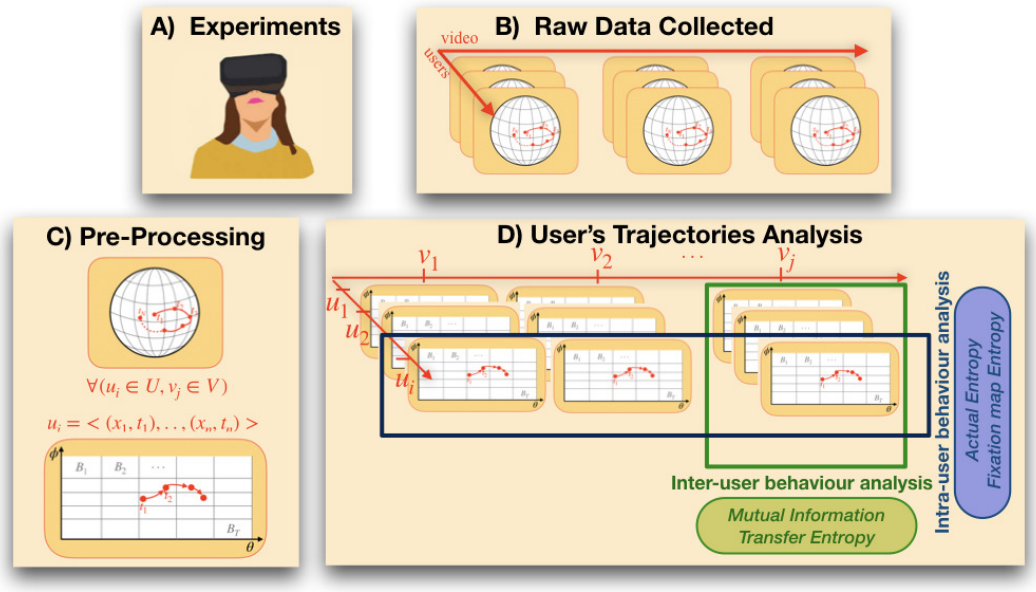


Figure 5.1: Overview of user behaviour analysis in a VR system: **A)** Collection of user's trajectories during immersive experiments. **B)** The raw data collected from different users and content are stored in a database. **C)** After a general pre-processing (i.e., re-sampling), the VR trajectories are transformed in the most suitable format for the final analysis. **D)** Information-theory metrics are applied to the VR trajectories looking for the desired characteristics: *intra-* and *inter-user behaviour analysis*.

5.2 User behaviour analysis in VR

Figure 5.1 summarises the framework considered in this work, and aimed at identifying key behavioural aspects during an immersive experience.

The first step of any experimental study is the data collection (Figure 5.1 A). In a VR scenario, the data is the set of *navigation trajectories* that identifies movements of users while experiencing an immersive content. As defined in Chapter 2.1, the 3-DoF VR user trajectory is formally denoted by $\{(x_1, t_1), (x_2, t_2), .., (x_n, t_n)\}$ where t_i is the data acquisition time (*i.e.*, video timestamp) with $t_i < t_{i+1}$, $\forall_{1 \leq i < n}$, while x_i represents the spatial coordinates of the viewing direction (corresponding to viewport's centre). Based on the selected convention, x_i can be recorded in different formats: quaternion, spherical coordinates and Euler angles are the most common representations in VR. As described in Section 3.2, nowadays many datasets containing these VR trajectories are already publicly available (Figure 5.1 B). In particular, they provide a collection of head and/or eye-gaze positions as proxy of the viewing direction for a set of users which explored different VR images/videos. Data collection campaign is beyond the scope of this chapter, which is mainly focused on the analysis of these data. Therefore in the following we use the $360°$ video dataset provided in [109]; further details on the database will be given in

Section 5.4.1.

From the collected raw data, some pre-processing is usually needed (Figure 5.1 C). In our case, users trajectories are stored as quaternion, and not at a constant sampling rate. Thus, we firstly re-sampled all the collected data based on the frame rate of the corresponding video. For the sake of notation, in the following we denote the VR trajectory by $\{x_1, x_2, x_n\}$ omitting the timestamp t_i. Then, we converted the original format data (*i.e.*, quaternion) in two different formats more suitable for a behavioural analysis, neglecting in this way the viewport's rotation which is already well-known to not be so relevant. As depicted at the top of Figure 5.1 C, the spatial position x_i is represented in spherical coordinates by latitude-longitude pair – *i.e.*, $x_t = (\theta_t, \phi_t)$ with $0 \leq \theta_t < 2\pi$ and $0 \leq \phi_t \leq \pi$. To be compliant with most of the behavioural analysis tools, we also quantized the spherical content into regular block, each one with an assigned ID value (*i.e.*, $B_1, B_2, .., B_T$ in Figure 5.1 C, lower part).

The data is then ready to be processed, Figure 5.1 D. This step is the core of our proposed VR analysis framework, and it is aimed at better understanding user's navigation within omnidirectional contents. The analysis highlights a two-line investigations:

a) **intra-user behaviour analysis** aims at characterizing the interaction of each user over time against different video contents. Studying single user's trajectory over time allows us to profile user or to identify recurrent navigation patterns.

b) **inter-user behaviour analysis** aims at studying a user behaviour in correlation with others. The target here is to understand how much user's trajectories are informative in understanding/predicting other user's behaviours.

For both directions, we propose to use IT metrics due to their powerful ability in quantifying interactions within the same or between different sources of information. In the following section, we present the metrics that we consider in our behavioural analysis in a VR scenario.

5.3 Information-theoretic metrics

Information theory has been introduced by Shannon in [175] to answer fundamental questions on communication theory. Since then Information-Theoretic metrics have been applied to a much wider range of disciplines beyond communications, becoming a de-facto statistical tool for data analysis in fields such as physics, computer science, and neuroscience [176]. A key quantity in information theory is *entropy*,

which relates to the *uncertainty* or *randomness* associated with an event. The less an event is certain, the more informative the event is, resulting in higher entropy. In other words, the entropy is a measure of information required on average to describe a random variable [174]. More formally, given a random variable X, with x being one possible realisation of X, the entropy is measured by:

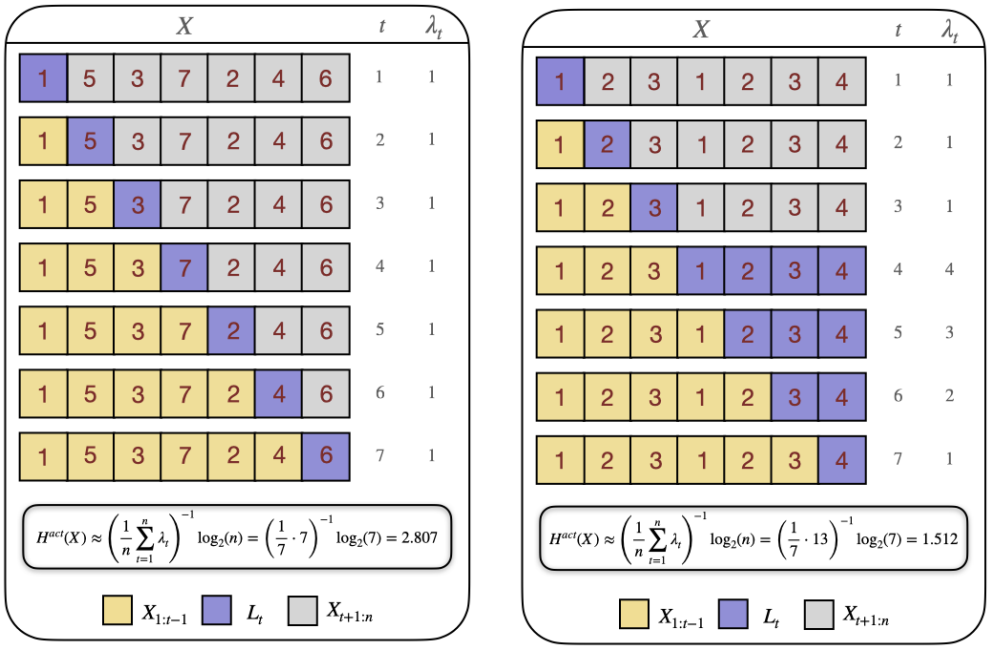
$$H(X) = -\sum_{x \in X} p(x) \log \big(p(x)\big) \tag{5.1}$$

where $p(x)$ is the probability of experiencing the event x. An event occurring with high-probability $p(x)$ is poorly informative (low entropy). Conversely, the occurrence of a very unlikely event carries a large information. The concept of information reflected by the entropy is highly related with the degree of *predictability* of a variable, with low values of entropy for highly predictable events. Authors in [173] exploited this correlation by using the entropy as a proxy of predictability of human mobility patterns. Specifically, they introduced the *actual entropy* to measure the information (and predictability) carried within a given trajectory, considering both the visiting rate but also the temporal order of visited areas. Specifically, the intuition is that if there are many repetitions (*i.e.*, same visited locations in the same order), the mobility trajectory results to be easy to predict [177]. In detail, the actual entropy can be estimated from the past history of user's trajectory by Lempel-Ziv compression algorithm [178]. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a trajectory of n points sampled at periodic time with x_t being the position at the t-th time-slot, and let $L_t = \{x_t, x_{t+1}, \ldots x_{(t-1)+\lambda_t}\}$ be a sub-sequence of X starting at time t and spanning λ_t time-slots, the actual entropy assumes the following form:

$$H^{act}(X) \approx \left(\frac{1}{n}\sum_{t=1}^{n} \lambda_t\right)^{-1} \log_2(n) \tag{5.2}$$

where λ_t is the length of the shortest sub-sequence in X starting at time-slot t and not appearing between time 1 and $t-1$. To be noted, the actual entropy of a given trajectory X is inversely proportional to the sum of the shortest non-repeated sub-sequences within it (*i.e.*, $\sum_{t=1}^{n} \lambda_t$).

To provide further intuition on the actual entropy, in the following we show how to estimate this entropy in two different scenarios: a trajectory highly random, $X = \{1, 5, 3, 7, 2, 4, 6\}$, and a trajectory which presents a repetition, $X = \{1, 2, 3, 1, 2, 3, 4\}$, respectively in Figure 5.2 (a) and Figure 5.2 (b). Each sub-figures shows the corresponding value of λ_t at each time-slot t. In Figure 5.2 (a), each location in X is a new one and therefore, contributes as 1 to each λ_t. This



(a) Highly random trajectory → high entropy. **(b)** Repeated sub-sequences → low entropy.

Figure 5.2: A visual example to evaluate $H^{act}(X)$ in two different scenarios: the sequence in **(a)** $X = \{1, 2, 3, 4, 5, 6, 7\}$ is highly random while the one in **(b)** $X = \{1, 2, 3, 1, 2, 3, 4\}$ presents a repetition of sub-sequences. The notation L_t represents the shortest sub-sequence in X starting at time-slot t and not appearing between time 1 and $t-1$ such that λ_t is equal to $|L_t|$.

behaviour brings to a high value of actual entropy, 2.807 (*i.e.*, low total sum of λ_t). On the contrary, the presence of a repeated sub-sequence in the example of Figure 5.2 (b) produce a more informative trajectory and thus, lower value of entropy which is equal to 1.512. Specifically, the maximum value of λ_t is at $t = 4$ where $L_t = \{1, 2, 3, 4\}$. In fact, while the values $\{1, 2, 3\}$ were previously present in the trajectory, the location 4 never appeared in X making this sub-sequence as the shortest sub-sequence starting at time $t = 4$ that does not appear before.

An other fundamental metric of information theory is the Mutual Information (MI). This metric measures the reduction of uncertainty of a random variable X provided by the knowledge of a second variable Y [174]. A large MI indicates that most of the information about X can be inferred from Y reducing therefore the uncertainty on X. Recalling the conditional entropy $H(X|Y)$ as the uncertainty of X given Y, the MI is defined for two variables X and Y as:

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) = \\ &= \sum_{x\in X, y\in Y} P(x,y) \log\left(\frac{P(x,y)}{P(x)P(y)}\right) \end{aligned} \tag{5.3}$$

Table 5.1: Key features of the video sequences analysed in this chapter.

ID	Name Video	Selected Segment	Description
01	Diving	00:40 - 01:40	No main FoA
02	Paris	00:00 - 01:00	Scene cuts with always one or more FoAs
03	Rollercoaster	01:05 - 02:05	One main FoA
04	Timelapse	00:00 - 01:00	moving FoAs distributed on the horizon line
05	Venice	00:00 - 01:00	No main FoA

where $p(x, y)$ is the joint probability of experiencing both events x and y, and $P(x)$, $P(y)$ their marginal distributions. To note that MI is zero if the two variables are uncorrelated, *i.e.*, $p(x, y) = p(x)p(y)$.

Finally, Transfer Entropy (TE) is a conditional entropy that considers not only the occurrence of events but also their temporal ordering. This metric measures the reduction of uncertainty about the future value of a variable (Y_{future}) by knowing the whole past history of itself (Y_{past}) and of a second variable (X_{past}). Therefore, TE is defined as follow:

$$\begin{aligned} TE(X \rightarrow Y) = \\ = H(Y_{future}|Y_{past}) - H(Y_{future}|X_{past}, Y_{past}). \end{aligned} \tag{5.4}$$

In contrast with MI, TE measures better the influence from X to Y.

5.4 Results

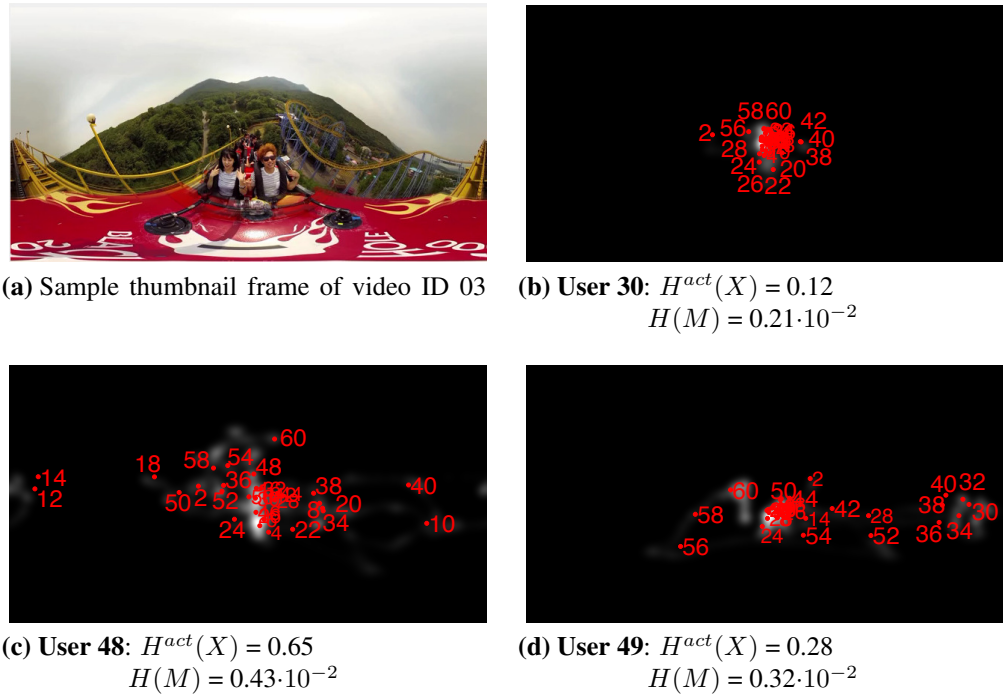
In the following, we first describe the dataset used in our VR behavioural analysis. We then provide and comment experimental results for both *intra-* and *inter-user behaviour analysis* (Figure 5.1 D). In more details, we adopt the metrics described in the previous sections, in the case of X and Y being users trajectories.

5.4.1 VR Trajectory Dataset

To apply our proposed framework of user behaviour analysis, we chose the dataset published by Corbillon *et al.* [109]. This dataset collects navigation trajectories of 57 users who navigated within 5 omnidirectional sequences. Table 5.1 shows a wide selection of content features of this dataset in terms of FoAs, scene cuts, etc..

5.4.2 Intra-User behaviour analysis

Our first direction of VR user analysis aims at characterising each user individually looking for patterns over time and across different contents. For example, some viewers could be generally interested in exploring the immersive video, independently from the content, and others be always static. This would indicate that the user's behaviour does not depend only on the content but it also influenced by the



(a) Sample thumbnail frame of video ID 03

(b) User 30: $H^{act}(X) = 0.12$
$H(M) = 0.21{\cdot}10^{-2}$

(c) User 48: $H^{act}(X) = 0.65$
$H(M) = 0.43{\cdot}10^{-2}$

(d) User 49: $H^{act}(X) = 0.28$
$H(M) = 0.32{\cdot}10^{-2}$

Figure 5.3: Examples of video ID 03 **(a)** of fixation maps for 3 different users **(b, c, d)**. In red, the fixation positions and the corresponding timestamp.

personal attitude of the viewer.

To study independently the behaviour of each user while navigating, we adopt the actual entropy ($H^{act}(X)$) which quantifies the similarities over time within the same variable. We compare the actual entropy with the entropy of fixation map, $H(M)$, evaluated as the entropy of all fixation points[1] per user for each video recorded during experiment. This metric is typically used to evaluate model of visual attention, and gives a qualitative idea about the dispersion of movements over time. In both metrics, a low value of entropy means that the user is focused on a restricted area; while high value stands for more exploratory movements. The main difference between the two metrics is that the actual entropy considers temporal order of navigation points which is neglected by the fixation map entropy.

Figure 5.3 shows one frame selected from video ID 03 (Figure 5.3 (a)) and fixation maps evaluated by three different users (users 30, 48 and, 49). Corresponding values of the entropy metrics are also provided in each subfigure caption (Figure 5.3 (b)–5.3 (d)). For user 30 both metrics are in agreement as they are both low. This is explained by the very focused fixation map shown in Figure 5.3 (b). Conversely, fixation maps of users 48 and 49 are more spread along the equatorial area (Figure 5.3 (c) and Figure 5.3 (d), respectively). This leads to higher values

[1] In this work, we consider user's head positions as proxy of their fixation points.

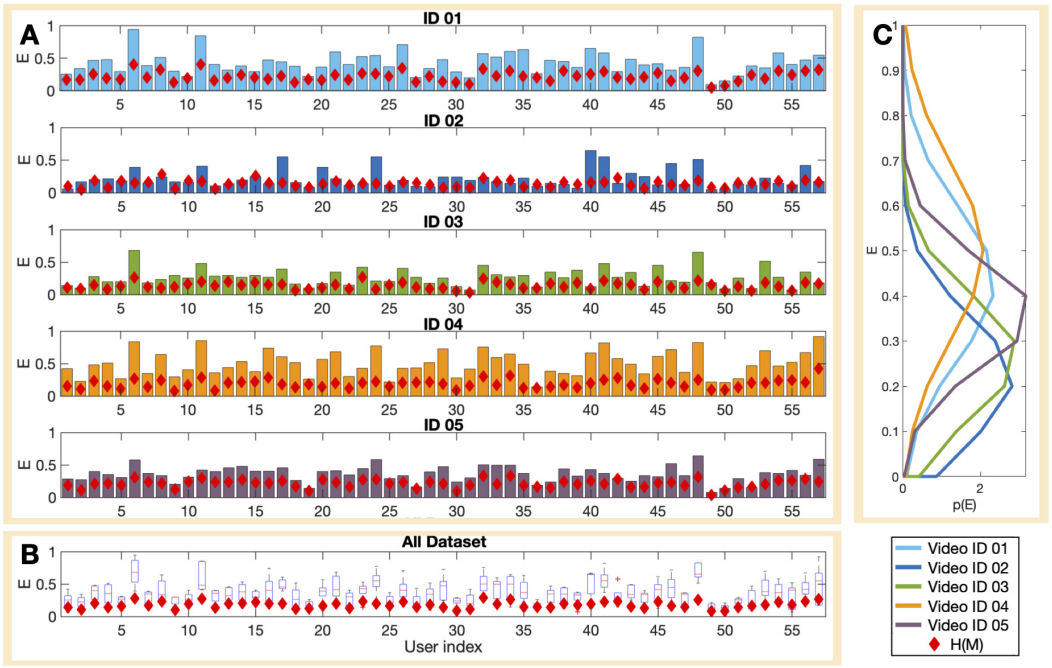


Figure 5.4: Intra-user behaviour analysis: **A** entropy of each user per video; **B** statistical analysis of the entropy for all users across the dataset; **C** probability distribution of actual entropy for each video across users.

of entropy, as already anticipated. Interestingly, there is a significant difference in terms of actual entropy for these last two viewers (0.65 for user 48 and 0.28 for user 49), difference that is not fully captured by the entropy of fixation map. Looking at the distribution of timestamps (*i.e.*, red numbers appearing in the fixation maps), we can notice that user 48 is navigating more randomly inside the content. User 49 is also moving within the content, but his/her fixation points are more contiguous over time. For instance, from time 30 to 40 the user remains in the right side of the panorama. Thus, actual entropy seems to detect discontinuity and randomness in the trajectories better than $H(M)$. Beyond the above visual results, Figure 5.4 provides a more exhaustive analysis of the actual entropy for the entire dataset. In particular, Figure 5.4 A depicts the actual entropy (bar plot), and the entropy of fixation map (red diamond) per user and per video. It is worth noting that most of the users preserve consistent behaviour across videos. Users with high value of actual entropy in a single video tend to experience high actual entropy also for other videos (see user 6); the same for small values of actual entropy (see user 50). This is a remarkable observation as it shows that users can be profiled across different videos. This is confirmed by the statistical analysis for all users across videos showed in Figure 5.4 B, which provides box plot of the actual entropy. The variance of the actual entropy is indeed kept small for the majority of the viewers. Finally, even if the content might not play the dominant role in defining user's behaviour,

it is still worth mentioning that it plays an important influence. Figure 5.4 C depicts the probability distribution of the actual entropy per video. This plot shows that video ID 02 (one main FoA) has the lowest mean value and small variance of actual entropy; conversely video with more FoAs, such as video ID 01 and 04, are characterised by higher mean value and variance of the metric. This means that the way in which users navigate within the omnidirectional content is more diversified when there are more FoAs.

5.4.3 Inter-User behaviour analysis

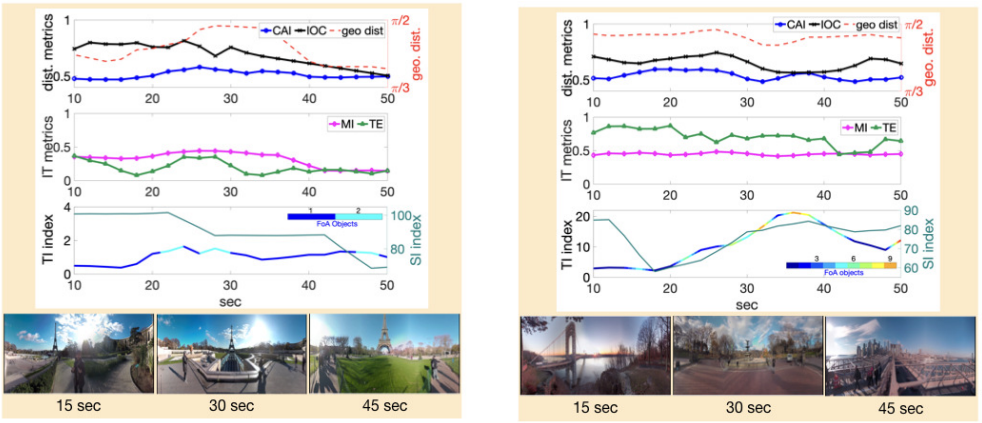
While the intra-user analysis provides a way to profile each user based on their way of navigating within VR contents, we are now interested in extending the behavioural analysis with a comparison among users. In particular, we aim at measuring key differences among navigation patterns of different viewers over time within the same content. To carry out this inter-user behaviour analysis, we use Mutual Information (MI) and Transfer Entropy (TE). As defined in Section 5.3, these two entropy-based metrics allow us a pairwise similarity analysis among users considering their positions over time (*i.e.*, their trajectories).

As benchmarking, we also analyse user's behaviour with two tools existing in the literature and based on the distance among users: Inter-Observer Coungrency (IOC), and clique-based clustering algorithm for VR trajectories. The first metric has been proposed in [179] as a measure of similarity for users viewing traditional images, and it is based on a one-to-all comparison (*i.e.*, the heatmap of a single user is compared against the one computed based on all other users). Instead, the clique-based clustering is defined in Chapter 4 and detects viewers that display similar viewports while consuming an immersive content. To quantify the consistency among users detected by the clique-based clustering, we define a Clique-Index (CI). Given the set of clusters at instant t, the CI for a user u is the number of users in the same cluster of u at time t (*i.e.*, $w_t(u)$) normalised per the size of the maximal cluster (*i.e.*, in terms of number of elements) at time t (*i.e.*, $w_t^{max} = \max_u w_t(u), \forall u$). More formally:

$$CI(u) = \frac{w_t(u)}{w_t^{max}}, \;\; \forall t = 1, ..., T \tag{5.5}$$

where T is the total length of the analysed video.

Equipped with the above notation, we can now provide the inter-user behaviour analysis. First, we split each content in temporal segments of duration 2 sec. (*i.e.*, typical chunk length in video streaming systems). Then, we compute per each segment MI and TE between users adopting the software provided by [176], and heatmaps by the tool presented in [180]. To preserve consistency, we compute



(a) Video ID 02 **(b)** Video ID 04

Figure 5.5: Inter-user behaviour analysis: top subplot shows distance metrics (*i.e.*, CAI and IOC), middle IT metrics (*i.e.*, MI and TE), bottom one content information (*i.e.*, TI, SI indexes and number of FoAs. The latter is reflected by the colour of the curve). At the bottom, there are 3 thumbnail frames corresponding to different temporal instant of the video.

clique-based clusters (*i.e.*, trajectory-based format) over a time-window of 2 sec.. Finally, to verify a correlation between user's movements and video content, we also evaluate content characteristics in terms of temporal and spatial information indexes (TI and SI, respectively [167]) and, number of main FoA objects detected in the scene by a Multiple Object Tracking tool [181].

For the sake of brevity, we focus on results carried out by only two videos, namely ID 02 and ID 04. In particular, these videos cover different content characteristics as shown in the frames provided at the bottom of Figure 5.5 (a) and Figure 5.5 (b). Specifically, video ID 02 has always one or two static main objects in the scene: a tour-guide and the Eiffel Tower in the first two frames; only the tower at the end of the video. On the contrary, video ID 04 is rather characterised by many fast-moving objects. These intuitions are confirmed by content information metrics (*i.e.*, TI, SI indexes and number of FoA detected objects) provided in the bottom subplot of Figure 5.5 (a) and 5.5 (b). The number of detected FoAs is identified by the colour code of the TI curve. Video ID 02 has only one or two FoA objects, and a TI index much lower than the one for video ID 04. Conversely, video ID 04 has more FoA objects with a peak of 9 around the middle of the sequence. The remaining subplots of Figure 5.5 show the inter-user metrics introduced in Section 5.3 as a function of time, and averaged across users. In the top subplot there are metrics based on spatial distance such as CI and IOC compared with the averaged pairwise geodesic distance between users over time (red dashed line). The middle subplot depicts instead the IT metrics MI and TE. The entropy-based metric TE seems to

reflect quite well the content information, especially the TI index and the number of FoA objects. In video ID 02 (Figure 5.5 (a)), TI increases around 20-30s and FoA objects are two instead of one. Users react by having a more exploratory trajectories – reflected by higher geodesic distance. This increase of randomness in the trajectories is measured well by the TE that peaks in this temporal range. Finally, video ID 04 has many more FoAs detected objects than ID 02. This leads to a more random navigation of viewers, proved by higher TE values. This difference is captured by TE but not by the spatial metrics, top subplot in both figure. Moreover, the TE range in video ID 02 is substantially lower than the one experienced in the other sequence (Figure 5.5 (b)), indicating a wider navigation of the scene in video 04. This preliminary study has shown a tight correlation between content information and TE.

5.5 Chapter Summary

In this chapter, we have proposed a novel methodology for behaviour analysis in a 3-DoF VR scenario aimed at characterising navigation patterns across content or across users. This is carried out by considering a space-time trajectory domain rather than only a spatial domain. Indeed, some users might be highly driven by the content when navigating, while others might be highly static or highly dynamic despite of the content. Or at the same time, some content features could be so dominant to led at very similar navigation trajectories among viewers. By leveraging on the knowledge from different disciplines, we based our behavioural investigation on *information-theoretic metrics*. The key intuition is to show that these IT metrics allow us to *quantify* the actual behaviour of users navigation. We conduced an *intra-user behavioural analysis* focused on understanding the behaviour of each individual when navigating in VR. By measuring the actual entropy of navigation trajectory, we identified for some users consistent patterns across different contents. For example, some users experience a more predictable trajectory for all videos. We also observed a correlation between content and actual entropy: the lack of a dominant FoA leads to more discontinuity and randomness in navigation trajectories. As second step, an *inter-user behavioural analysis* was carried out, aimed at understanding how much information about a single content can be extracted when observing an entire population of viewers. The transfer entropy showed to better quantify behavioural similarity among users rather than the metrics based on spatial distribution.

Both this chapter and the previous one contributed to the study of interactive users and to the development of new behavioural analysis tools and methodologies,

built specifically for 3-DoF immersive environments. Identifying similarities in the navigation is indeed a step forward in modelling how users behave in virtual environments and it is a key factor to better optimise experiences around the users. To show the impact and advantages of taking into account user behaviour in immersive systems, in the following chapter we formulate a first example of optimal user-centric solutions.

Part III

USER-CENTRIC 3-DoF VR SYSTEM

Chapter 6

Investigation of Users Influence on the System Design

In this part of the thesis, we show the importance of considering users behaviour when designing 3-DoF VR streaming systems. We present two case studies of user-centric solutions: the one presented in this current chapter aimed at optimising the ODV encoding and storage at the main server; and the one detailed in Chapter 7 presenting an optimal transmission strategy for VR applications capable of satisfying bandwidth requirements while optimising the quality of the end-user experience in navigation.

6.1 Introduction

To unlock the great potential of Virtual Reality applications in their online format (*e.g.*, social VR, e-learning, virtual training), there is the need to develop VR communication platforms able to sustain the high-load and low-latency requirements. The envisioned solution to this is to build personalised (or user-centric) systems, which put the user at the center of the whole coding/delivery/rendering chain. Due to the increasing cost of storage and coding, optimising the storage space at the main server has become also a fundamental need, especially for VR content − highly data intensive. Works focused on server optimisation for classical adaptive streaming platforms have been already proposed in literature [83], tuning the coding rate and resolution depending on both the population features and the type of content. In the context of ODV, only [84] introduces a content-aware encoding ladder estimation that achieves cost-optimal and higher objective quality compared to recommended encoding ladders. However, information about users navigation within the content is not considered. Hence, to carry out a case study in this chapter we also bring-in the novelty of formulating a *user-centric server optimisation* for ODV adaptive streaming systems. In particular, we evaluate the optimal set of coding parameters

to store ODVs at the main server minimising the total cost and maximising user's experience, taking into account the users' behaviour and network characteristics. Results show that our solution performs well in terms of total cost (*i.e.*, encoding and storage cost) and quality experienced by users. Most importantly, results reveal also a correlation between the optimal set and similarity (*i.e.*, affinity) in users navigation.

This chapter contributes to the overall open problem of optimally designing a VR system, with a case study of 3-DoF VR system optimised from the server perspective, with a two-folds novelty: *i)* the proposed problem formulation; *ii)* the translation of the users' behaviour analysis into gain for a system provider. We verify and test the proposed optimisation problem on our novel 3-DoF navigation dataset presented in Chapter 4 (Section 4.5). As a reminder, the main novelty of this dataset is in tracking and collecting users navigation across three different VR devices (*i.e.*, HMD, laptop and tablet) and in analysing the acquired data with new user similarity metrics (*i.e.*, User Affinity Index (UAI)). The remainder of this chapter is organised as follows. Related works on streaming strategies in VR system are reported in Section 6.2. The case study is formulated in Section 6.3. Section 6.4 and Section 6.5 describe metrics and simulation settings, respectively. In Section 6.6, the performance of the proposed optimisation algorithm is first compared with the set of recommended representations and then, the results are further analysed to reveal the effect of the user behaviour. Finally, the chapter is summarised in Section 6.7.

6.2 Related Works

Although streaming strategies have been widely investigated in recent decades, many open challenges are still unsolved in the context of user-centric immersive communications. We now describe the latest contributions mostly related to our work, which is focused on user-depended streaming strategies for ODV. For a comprehensive literature review on ODV analysis and communication, we refer the reader to Chapters 2 and 3 of this thesis.

In recent years, user-centric systems have been developed, optimising every step of the ODV video delivery chain: coding [182], streaming [87], caching [183, 184], and rendering [185]. In particular, tile-based coding systems [182, 186] were utilised using viewport adaptive streaming algorithms [137, 187, 188] to provide smooth VR video experience [87]. For instance, Nguyen *et al.* [137] presented an adaptation logic for ODV streaming to decide an optimal version of each tile according to users head movements and network bandwidth. Their analysis empha-

sised the need of accurately predicting future viewport position for ODV streaming. In the aspect of the prediction of future viewport, Petrangeli *et al.* [130] proposed a prediction algorithm for long-term prediction of user viewport. In their work, the navigation trajectories of a given user are modelled over time such that future viewports can be predicted based on the navigation patterns of users stored in the system. According to their results, their proposed algorithm can increase prediction accuracy of the expected viewport area by 13% on average compared to previous algorithms. By looking more at the server-side (i.e., coding optimisation), Ozcinar *et al.* [87] proposed a visual attention-based ODV streaming system optimising the tile-based design taking into account users saliency maps. The work showed the importance of being user-centric also at encoding side without focusing a design of cost-aware VR system. In contrast, Xiao *et al.* [189] optimised the tile-based encoding design of ODVs seeking the best trade-off between storage costs and overall quality of the panorama. However, the storage cost was not formally optimised and the users trajectories were neglected in the problem formulation. There are also some activities in the sense of standardisation bodies, such as MPEG-I [190]. For instance, a practical study by Graf *et al.* [64] examined several adaptive streaming strategies and evaluated bitrate overhead with quality requirements in VR. To find the optimal set of quality-variable video versions for ODV streaming, Corbillon *et al.* [187] presented an optimisation model for the concept of quality regions of ODVs. Their main contribution is to consider the surface bitrate and users head movement data within the proposed optimisation framework. However, their study was restricted to using the concept of quality-emphasised regions, with the employed constraints being the number of quality-variable video versions and the bandwidth. Also, Zou *et al.* [191] proposed a server-side rate adaptation problem for the tile-based adaptive ODV streaming. They aimed to maximise the QoE of multiple users who are competing for transmission resources at the network bottleneck. Furthermore, Chakareski *et al.* [192] maximised the QoE for given network resources at the server side. Their work consider user navigation trajectories and spatio-temporal rate-distortion characteristics of a given video. However, the proposed formulation is based on the traditional Mean Square Error (MSE), which does not take the spherical distortion of ODV representation into account. In summary, from the literature it is clear the importance and the gain in being *user-*, *cost-*, and *geometry-aware* when designing VR systems. However, such a complete design at the server side is missing.

Our work goes beyond the state-of-the-art as we take into account our users behaviour analysis, formulating a novel user-centric server optimisation system, which minimise the user-centric spherical quality and the coding and storage costs.

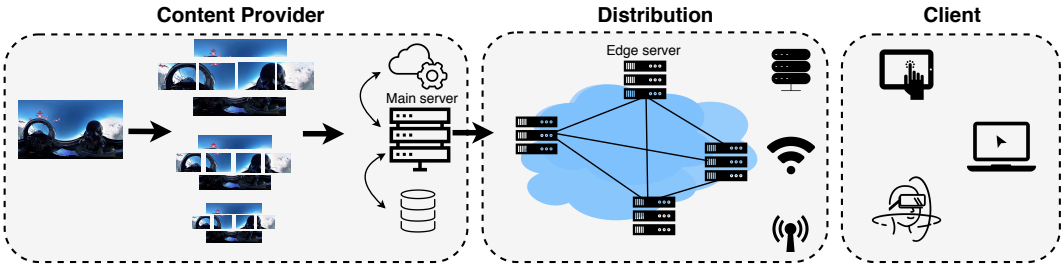


Figure 6.1: Schematic of the adopted tile-based adaptive ODV streaming system.

In particular, we developed an optimisation algorithm to determine the optimal set of coding parameters to store ODVs at the server minimising the total cost and maximising users experience. Differently from the aforementioned works, the main novelty of our algorithm is to take into consideration users behaviour beyond the spherical geometry and content information, minimising the total cost and yet maximising the final quality for ODV adaptive streaming systems. A further novelty is to link the optimal design with the affinity of users navigation patterns.

6.3 User-centric Server Optimisation

We now show the importance of considering users behaviour when designing an ODV streaming system defining a *user-centric server optimisation* that considers multiple VR devices. In particular, we focus on optimising the set of tile-representations to store at the server, considering spherical geometry, content complexity of ODVs and network capacity beyond users navigation features. First, we introduce the system model for the tile-based adaptive ODV streaming scenario adopted in this work. Then, we formulate an Integer Linear Programming (ILP) used to evaluate the optimal set of tile-representations that maximises the quality perceived by users while minimises the total cost of encoding and storage.

6.3.1 System model

Figure 6.1 illustrates the adopted tile-based adaptive ODV streaming system. Namely, each video sequence is spatially decomposed into tiles, which are encoded at different coding rates and resolutions. The generated representations of each tile are then temporally segmented into chunks of fixed duration (*i.e.*, typically 2 *sec.*) and stored at the main server. Out of the many representations stored at the server, only one per tile is actually distributed through edge servers to the final user. The selection of the representation is usually performed at the client side. Specifically, any final users, while navigating inside an ODV, will periodically requests to download the most suitable set of tile-representations (*i.e.*, such that to cover the entire panorama), based on the available bandwidth and their current position inside the ODV – usually the best quality that meets bandwidth constraints. In particular, we

consider users downloading the entire panorama at each downloading opportunity but at heterogeneity quality levels. Specifically, the more probable tile is the higher quality at which it is downloaded. In this contest, we are interested in investigating how to design an optimal representations set at the server side able to satisfy the requests from a potential VR population.

More formally, let $\mathcal{V}$ be the set of ODVs available at the *main server*. Each video $v \in \mathcal{V}$ is decomposed into N tiles. We denote by $j \in \mathcal{J}_v = 1, 2, .., N$ the set of tiles belonging to v. Then, each tile is encoded independently into different representations characterised by bitrate levels, $r \in \mathcal{R}$ and, spatial resolutions $s \in \mathcal{S}$. Note that $\mathcal{R}$ and $\mathcal{S}$ are sets of admissible bitrates and spatial resolution values. All variables v, j, r and s are integer values that represent the index in their corresponding set. In particular, the nominal value (in *kbps*) of the encoding rate r is denote by b_r and $\mathcal{B}$ is the set of available bitrates. Each representation is temporally divided into chunks of a fixed duration. Let $\mathcal{L}^v = \{(j, r, s) | j \in \mathcal{J}_v, r \in \mathcal{R}, s \in \mathcal{S}\}$ be the set of *representations* per chunk of a video $v \in \mathcal{V}$; the triple (j, r, s) indicates the representation of tile j encoded at bitrate r and resolution s. Given the heterogeneity of users downloading ODV (*i.e.*, different type of network and devices), all the possible representations $\bigcup_v \mathcal{L}_v$ should be stored at the main server. This would ensure to serve each users' request at the best. In practice, coding and storage costs can be unbearable when all representations are stored. Hence, the need to select a subset of representations $\mathcal{L} \subseteq \bigcup_v \mathcal{L}_v$ to store at the main server. Our goal is then to seek the optimal subset $\mathcal{T}^*$ to be available at the server in order to maximise the QoE given constraints from both the server and client perspectives. We argue that in this system design optimisation, the knowledge of displaying device and video category as well as the user navigation trajectories is the key for any efficient optimal set.

Let $\mathcal{U}$ be the set of all clients served in our ODV streaming system. We assume that all final users can be categorised based on the selected video content, viewing device and the kind of network connection (*i.e.*, capacity of each user connection). Namely, a user of type $u \in \mathcal{U}$ is defined by the desired video $v^u \in \mathcal{V}$ displayed at the resolution of the selected device $m^u \in \mathcal{M}$, downloaded based on the kind of network $n^u \in \mathcal{N}$. Without loss of generality, we make the assumption that each device is associated with a single display resolution. The type of network n selected by user u defines the range of available throughput value BW^u. Finally, each type of users has an own navigation path inside the ODV, that depends on the selected device d^u as well as the content and the user itself. Therefore, we define p_j^u as the probability of tile j to be displayed by user's category u. Finally, we denote by δ^u the portion of users of type-u, with $\sum_{u \in \mathcal{U}} \delta^u = 1$.

Table 6.1: Notation adopted in the problem formulation.

Name	Description
$\mathcal{U}, u \in \mathcal{U}$	set of all users' type and the actual user served in the system, respectively
$\mathcal{V}, v^u \in \mathcal{V}$	set of ODV and video content requested by user u, respectively
$\mathcal{J}_v, j \in \mathcal{J}_v$	set of all tiles of the video v and the selected j tile, respectively
$\mathcal{R}, r \in \mathcal{R}$	set of all possible coding rate and the actual coding rate at which a tile can be encoded, respectively
$\mathcal{B}, b_r \in \mathcal{B}$	set of all available values of encoding rate and the nominal value of r (kbps),
$\mathcal{S}, s \in \mathcal{S}$	set of possible spatial resolution and actual spatial resolution s at which a tile j can be encoded,
(j, r, s)	representation of a tile j encoded at rate r and spatial resolution s,
$\mathcal{L}^v$	set of all possible tile-representations for a video v,
$\mathcal{T}^* \subseteq \mathcal{L}$	optimal set of representations stored at the main server,
$\mathcal{D}, d^u \in \mathcal{D}$	set of available device and actual device selected by user u
$\mathcal{S}, s^u \in \mathcal{S}$	set of available spatial resolution (i.e., screen size) and actual resolution of device selected by user u
$\mathcal{N}, n^u \in \mathcal{N}$	set of available networks and actual network selected by user u, respectively
BW^u	available bandwidth throughput for user u,
p_j^u	probability of tile j to be displayed by users of type-u,
δ^u	portion of users of type-u,
$D_j(r, s)$	distortion value of tile j encoded at rate r and resolution s
$C_j^{TOT}(r, s)$	total costs (encoding and storage costs) for a tile-representation encoded at rate r and resolution s

Table 6.1 summarise the main notations adopted so far and in the following problem formulation.

6.3.2 Problem Formulation

Given the set $\mathcal{L}$ of all possible representations for all videos $v \in \mathcal{V}$, we seek the optimal subset of representations $\mathcal{T}^* \subseteq \mathcal{L}$, which maximises the perceived quality during the navigation, minimises the total price of storage and encoding for the selected tile-set and yet a bandwidth constraint is respected. Our *user-centric server optimisation* problem can be defined as follows:

$$\begin{aligned} \mathcal{T}^* : \arg\min_{\mathcal{T}} & \sum_{u \in \mathcal{U}} D^u(\mathcal{T}) + \lambda C^{TOT}(\mathcal{T}) \\ \text{s.t.} & \sum_{(j,r,s) \in \mathcal{T}} b_r \leq BW^u \qquad \forall u \end{aligned} \tag{6.1}$$

where $D^u(\mathcal{T})$ is the spherical distortion experienced by type-u user achieved when the $\mathcal{T}$ representation set is available at the main server, λ is the regularisation term and $C^{TOT}(\mathcal{T})$ is the total cost to store and code $\mathcal{T}$. In particular, the distortion $D^u(\mathcal{T})$ is defined as follows:

$$D^u(\mathcal{T}) = \sum_{(r,s) \in \mathcal{T}} D^u(r, s) = \sum_{(j,r,s) \in \mathcal{T}} D_j(r, s) \hat{S}_j p_j^u \tag{6.2}$$

where j is a generic tile on the planar encoded at *r*-th rate and s-th resolution. To take into account the spherical geometry, the spherical distortion is weighted by $\hat{S}_j$ that is the normalised portion of the sphere covered by tile j (more details are provided in Section 6.4.1). Finally, p_j^u is the probability for the tile j to be displayed by a type-u user. Storing the video on the main server provider has a cost ($), which depends on both the content complexity (affecting the total file size) and the resolution of representations. We estimate this total cost $C^{TOT}(\mathcal{T})$ in Equation (6.1) as sum of the cost per each encoded tile ($\sum C_j^{TOT}(r,s)$). Since no prior assumption on distortion function (such as linear, quadratic, or convex function) is imposed, we preserve a general solving method and we cast the optimisation problem presented in Equation (6.1) as ILP problem introducing the following binary decision variables:

$$\begin{aligned} \alpha_{j,r,s}^u &= \begin{cases} 1, \text{ if user } u \text{ requests the representation } (j,r,s) \\ 0, \text{ otherwise} \end{cases} \\ \beta_{j,r,s} &= \begin{cases} 1, \text{ if any user request a representation } (j,r,s) \\ 0, \text{ otherwise.} \end{cases} \end{aligned} \tag{6.3}$$

Without loss of generality, we suppose that each user can request only tile-representation encoded at resolution s corresponding to the display resolution (*i.e.*, spatial resolution at which the content will be displayed) of the selected device m^u. Therefore, we also define the following auxiliary variable:

$$\gamma_s^u = \begin{cases} 1, \text{ if user } u \text{ requests representations at resolution } s \\ 0, \text{ otherwise.} \end{cases} \tag{6.4}$$

This leads to the problem formulation shown in Table 6.2 equivalent to the problem showed in Equation (6.1). The constraints (6.5a)-(6.5c) set up a consistent relation between the two decision variables. The constraints (6.5d)-(6.5f) makes homogeneous the resolution constraint by auxiliary variable γ. The constraint (6.5g) imposes bandwidth constraints. Finally, constraints (6.5h)-(6.5j) limit the decision variables to binary values.

The optimal solution of the ILP problem proposed in Table 6.2 is NP-hard and it can be evaluated by a generic solver IBM ILOG CPLEX [193] using a branch-and-cut algorithm. The method of branch-and-cut consists of a search tree technique with the application of cuts of the nodes in the tree. In particular, each node represents a LP sub-problem to be solved, and the creation of two new nodes from a parent

Table 6.2: ILP problem formulation for a user-centric optimisation.

Integer Linear Programming

$$\min_{\alpha,\beta,\gamma} \sum_{u\in\mathcal{U}} \sum_{j\in\mathcal{J}_v} \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} D_j(r,s)\hat{S}_j p_j^u \alpha_{jrs}^u + \lambda \sum_{j\in\mathcal{J}_v} \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} \beta_{jrs} C_j^{TOT}(r,s) \tag{6.5}$$

$$\text{s.t.} \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} \alpha_{jrs}^u \leq 1 \qquad \forall u, j \tag{6.5a}$$

$$\alpha_{jrs}^u \leq \beta_{jrs} \qquad \forall u, j, r, s \tag{6.5b}$$

$$\beta_{jrs} \leq \sum_{u\in\mathcal{U}} \alpha_{jrs}^u \qquad \forall j, r, s \tag{6.5c}$$

$$\sum_{s\in\mathcal{S}} \gamma_s^u \leq 1 \qquad \forall u \tag{6.5d}$$

$$\alpha_{jrs}^u \leq \gamma_s^u \qquad \forall u, j, r, s \tag{6.5e}$$

$$\gamma_s^u \leq \sum_{j\in\mathcal{J}_v} \sum_{r\in\mathcal{R}} \alpha_{jrs}^u \qquad \forall u, s \tag{6.5f}$$

$$\sum_{j\in\mathcal{J}_v} \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} \alpha_{jrs}^u b_r \leq BW^u \qquad \forall u \tag{6.5g}$$

$$\alpha_{jrs}^u \in \{0,1\} \qquad \forall u, j, r, s \tag{6.5h}$$

$$\beta_{jrs} \in \{0,1\} \qquad \forall j, r, s \tag{6.5i}$$

$$\gamma_s^u \in \{0,1\} \qquad \forall u, s \tag{6.5j}$$

node is a branch. It is worth mentioning that the branch-and-cut algorithm generally requires exponential computational complexity $\mathcal{O}(2^E)$ to achieve the optimal solution, with E being the cardinality of decision variables. In our case with the binary decision variables α, β and γ, we obtain E$\sim |\mathcal{U}|^2|\mathcal{J}_v|^2\mathcal{R}|^2|\mathcal{S}|^3$.

6.4 Metrics and user population

We now describe the objective functions used in this work to validate the optimisation problem proposed in the previous Section. First, we present the distortion

function and cost models that we consider to minimise storage capacity utilisation, ensuring a high quality of experience. Then, we define the different types of user population that reflect a wide set of clients in our simulated ODV adaptive streaming scenario.

6.4.1 Distortion model

Recalling Equation (6.2), we assume that the distortion $D^u(\mathcal{T})$ experienced by a generic type-u user is a popularity-weighted geometry-based distortion of a given set of tiles $\mathcal{T}$. In particular, this metric is *popularity-weighted* because we consider the probability p_j^u that a user of type u displays a specific tile j, while *geometry-based* because we introduce a scalar factor $\hat{S}_j$. In fact, due to the projection from spherical to planar domain, tiles have unequal sizes on the viewing sphere. To take into account this inconsistency, we introduce $\hat{S}_j$ which is defined in the following. Let us denote by $\mathcal{S}_j$, the surface on the sphere of the j-th tile centred in (θ_j, ϕ_j) and with $\Delta\theta_j$ and $\Delta\phi_j$ as longitudinal and latitudinal dimensions, given by:

$$\mathcal{S}_j = \int_{\theta_j - \frac{\Delta\theta_j}{2}}^{\theta_j + \frac{\Delta\theta_j}{2}} \int_{\phi_j - \frac{\Delta\phi_j}{2}}^{\phi_j + \frac{\Delta\phi_j}{2}} \sin\phi d\theta d\phi, \tag{6.6}$$

we therefore define $\hat{S}_j$ as follows:

$$\hat{S}_j = \frac{\mathcal{S}_j}{\mathcal{A}} \tag{6.7}$$

where $\mathcal{A}$ is the entire surface of the viewing sphere. In this framework, we assume a viewing sphere with ray unitary such that $\mathcal{A}$ is equal to 4π. Finally, the representation of a generic tile j encoded at rate r and spatial resolution s will lead to a distortion averaged over the entire tile denoted by $D_j(r, s)$ which can be evaluated on the planar format. In particular in this work, we adopt the Weighted MSE (WMSE) metric [194] as a distortion measure to compute $D_j(r, s)$ because of its pixel-based distortion estimation and low computational complexity. In detail, given a frame with resolution $W \times H$, the WMSE is defined as following:

$$\text{WMSE}(k, l) = \sum_{k=0}^{W-1} \sum_{l=0}^{H-1} (x(k, l) - y(k, l))^2 w(k, l) \tag{6.8}$$

where $x(k, l)$ and $y(k, l)$ are intensity values at the pixel position (k, l) for the reference and projected image, respectively. Instead, $w(k, l)$ represents the non-linear weights that takes into account the spherical geometry to Mean Square Error (MSE). Namely, this constant reflects the stretching ratio for pixel in position (k, l)

and depends on the planar-to-spherical projection. In this framework we consider Equirectangular Projection (ERP), hence each pixel weight is defined as follows:

$$w(k,l) = \frac{W(k,l)}{\sum_{k=0}^{W-1}\sum_{l=0}^{H-1} W(k,l)} \tag{6.9}$$

where $W(k,l)$ is the area scaling factor from equirectangular to unit spherical surface and is given by $W(k,l) = cos\Big[\Big(l - \frac{H}{2} + \frac{1}{2}\Big)\frac{\pi}{H}\Big]$.

6.4.2 Cost model

Beyond the distortion, another important aspect that the system designer should aim to minimise is the storage and encoding costs. Storing video representations at server providers (*e.g.*, Amazon, Microsoft, etc.) has a price that depends on the total size of the representations (in terms of *kbps*), and while storage cost might seems negligible, it is not when scaled for the number of video contents and representations that a content provider should have. Hence, there is a need for the proposed optimisation algorithm. Formally, the storage and coding costs is a function of the video complexity, resolution and encoding rate and it is defined as (cost per tile-representation) [84]:

$$C^{TOT}(\mathcal{T}) = C^{e}(\mathcal{T}) + C^{s}(\mathcal{T}) \tag{6.10}$$

where C^e and C^s are the encoding and storage costs, respectively. In particular, C^e is defined per each representation set $(\mathcal{T})$ as:

$$C^{e}(\mathcal{T}) = \begin{cases} \mu_e, & \text{if } s \leq 720p \\ 2\mu_e, & \text{if } 720p < s \leq 1080p \\ 4\mu_e, & \text{if } 1080p < s \leq 4\text{K} \end{cases} \tag{6.11}$$

where μ_e (\$) is a constant defined by the service provider and s is the resolution of each representation in $\mathcal{T}$. Instead, C^s is modelled as a linear function of the representation bitrate:

$$C^{s}(\mathcal{T}) = \mu_s \sum_{(j,r,s)\in\mathcal{T}} b_r \tag{6.12}$$

where μ_s (\$/GB) is a constant defined by service provider and b_r is the bitrate of the selected representation set $(\mathcal{T})$.

In our simulation settings, we follow the price-table of a real service provider [195, 196]. Therefore, we set $\mu_e = 0.1904$ \$/minute as the price to convert a video with an optimised quality in HEVC with frame rate ≤ 30 *fps* and $\mu_s = 0.024$ \$/GB. Both

costs refer to the area of Europe (London) in [195, 196].

6.4.3 Users population features

In a practical adaptive ODV systems, content providers serve a vast number of highly heterogeneous users. For the optimisation purposes, we categorise them based on key features. As defined in Section 6.3.2, a user $u \in \mathcal{U}$ is characterised by three parameters: requested video, viewing device and network type. Each of these parameters is modelled as follows.

- *Requested video content,* v^u. We consider the dataset of 15 ODVs presented in Chapter 4 (Section 4.5) composed by 3 different categories (*Documentary*, *Action* and *Movie*) with 5 video per category. We suppose that users can select each available video with the same probability (1 out of 15).

- *Selected rendering device,* m^u. Each user can display the video content on 3 viewing platforms (*i.e.*, HMD, tablet and laptop). Without loss of generality, we assume that users select device with equal probability (1 out of 3).

- *Type of network and related available bandwidth,* n^u *and* BW^u. We consider 3 types of networks (*i.e.*, 4G, WiFi, and ADSL) with their specific range of throughput and probability of experiencing that connection. For each type of connection, 3 different kinds of users have been considered, which means 3 values of bandwidth BW^u is possible per connection. Further details have been provided in the following.

In summary, we consider 27 types of users per video (3 types of devices $\times$ 3 types of possible networks $\times$ 3 possible bandwidth values). This ensures that our proposed *user-centric server optimisation* algorithm is tested under realistic settings with a complete and exhaustive set of clients, while preserving a limited complexity of the ILP problem.

Type of network and available bandwidth: Before presenting simulation settings and results, we complete the information about the user population clarifying the types of networks and available bandwidth. We consider 3 types of networks with their specific range of throughput, provided in Table 6.3. We assume that the probability of experiencing a given connectivity is linked to the device, as reported in Table 6.4. For each connection type, 3 different kinds of users have been considered: *i)* clients with bandwidth BW^u set as the 25-th percentile of the available bandwidth for the selected network, *ii)* users with bandwidth BW^u set as the 75-th percentile of the available bandwidth for the selected network, and *iii)* clients with

bandwidth BW^u set to the 50-th percentile of the available bandwidth for the selected network. We assume a probability 1/4 for a user to experience the first two cases and 1/2 to select the third downloading.

Table 6.3: Networks Bandwidth ranges.

Network Type	Minimum Bandwidth (Mbps)	Maximum Bandwidth (Mbps)
4G	4	20
WiFi	2	30
ADSL	5	35

Table 6.4: Probability of each network and device in our simulations.

Network Type	HMD	Tablet	Laptop
4G	0	0.6	0
WiFi	0.8	0.4	0.45
ADSL	0.2	0	0.55

6.5 Simulation settings

In this section, we provide the remaining details of the framework that we used to validate the proposed *user-centric server optimisation*.

6.5.1 Tiling and encoding

Each ODV was partitioned into six self-decodable tiles to deliver and render ODVs efficiently. Following the usual assumption of lower importance and low-motion characteristics of the poles and the dominant viewing adjacency of the equator [156, 168], we separate each ODV frame horizontally into three parts: one equator and two poles. The equator represents the middle segment, and the two poles stand for the top and the bottom sections of the frame. The size of equator is the double size of each pole. As the poles occupy the largest regions of the redundant pixels, in those areas, larger tile resolution size was used to compress them efficiently [156]. On the contrary, since the equator region contains the most dominant viewing probability, it is further divided vertically into 4 tiles to efficiently utilised them at both the server and client sides. Figure 6.2 illustrates the used structure for partitioning into self-decodable tiles and the tile index order that will be considered in the following.

We used the HEVC standard [197] to encode each tile of a given ODV. For this purpose, the *libx265* codec in the FFmpeg software (*ver.* N-85291) [198] was used. As recommended in [199], each tile was encoded using two-pass with 150 percent constrained variable bitrate configurations to ensure smooth video quality frame by frame for a wide range of devices. Before encoding, we scaled each video at different resolutions, $\mathcal{S} = \{1280 \times 720, 1920 \times 1080, 2560 \times 1440\}$. For the former one, as the content is already in the 2560×1440 resolution, no scaling was applied, and the two other resolutions were obtained by down sampling using the bi-cubic scaling technique. Here, we ensured that there is a noticeable

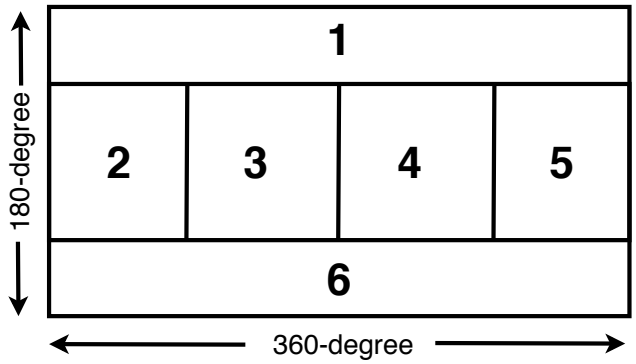


Figure 6.2: The used structure for tiling with tile IDs.

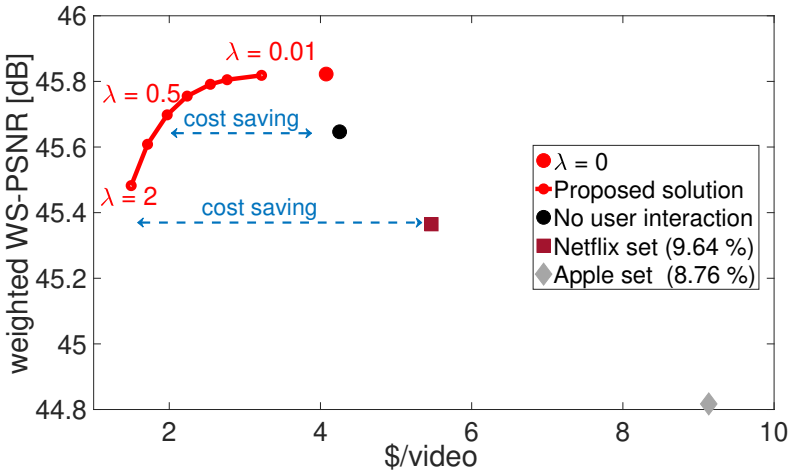
objective quality difference between each selection per ODV. Each scaled version of ODV was tiled and encoded using a set of target bitrate parameters $\mathcal{B}$ = $\{500, 760, 1005, 1529, 2326, 3537\}$ (in terms of *Kbps*). Each bit-stream was then divided into 2 *sec.* streaming chunks to perform adaptive streaming.

6.5.2 Comparative Methods

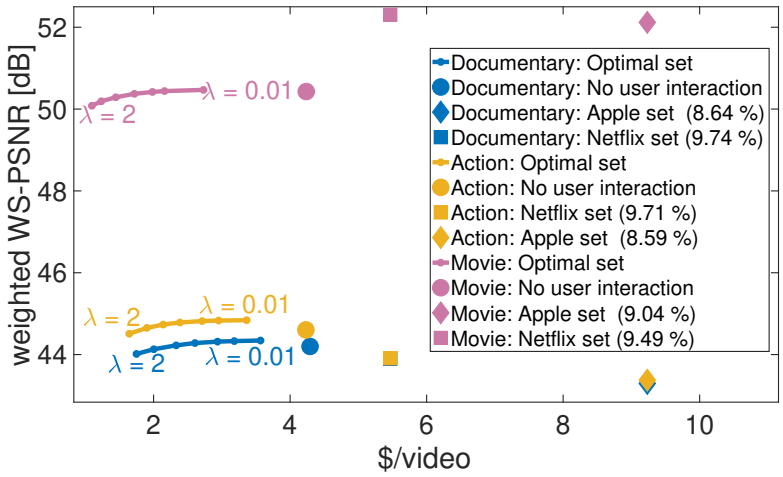
As last step of the simulation settings, we describe the benchmarking solutions for the optimisation server design. In particular, we evaluate the optimal sets of tile-representation with our user-centric algorithm (named "Optimal set" in the following plots) imposing different values of the regularisation parameter λ. In particular, we set $\lambda = [0.01, 0.05, 0.1, 0.25, 0.5, 1, 2]$. Then, we compare the performance of our optimisation with two sub-optimal solutions (*i.e.*, "$\lambda = 0$" and "optimal set - no interactivity") and two traditional recommendations sets (*i.e.*, "Netflix set" and "Apple set") [199, 200], which were originally developed for traditional 2D videos. "$\lambda = 0$" indicates the solution of our problem but neglecting the optimisation of costs, while "optimal set - no interactivity" omits also the probability p_j^u that defines where users most likely will focus their attention. The recommended bitrate sets of Apple and Netflix are defined as following: *i)* $\mathcal{B}$ = $\{400, 480, 560, 640, 750, 900, 970, 1170, 1350, 1670\}$ *Kbps* for the Apple set with corresponding encoded resolutions $\mathcal{S}$ = $\{720p, 720p, 720p, 720p, 1440p, 1440p, 1440p, 1440p, 1080p, 1080p\}$ and *ii)* for the Netflix set $\mathcal{B}$ = $\{390, 500, 720, 980, 1300, 1920\}$ *Kbps* and encoding resolution $\mathcal{S}$ = $\{720p, 720p, 1440p, 1440p, 1080p, 1080p\}$.

6.6 Simulation Results

The key goals of the proposed optimisation problem are $i)$ to ensure a good navigation experience within an ODV, reducing the total cost of encoding and storage; $ii)$ to show the advantage of taking into account users' behaviour in this optimisation. Figure 6.3 depict the averaged quality experienced by users (in terms of Weighted Spherical PSNR (WS-PSNR) [194]) as a function of the total cost, for



(a) Performance averaged per entire database



(b) Performance averaged per video category

Figure 6.3: Average experienced quality versus total cost of storage. In the legend on bracket, utilisation rate for non-optimal solutions.

the proposed optimal set representations as well as the benchmark ones introduced in Section 6.5.2. The experienced quality has been evaluated as the average quality of each tile weighted by its probability of being displayed in a specific scenario (*i.e.*, selected video and viewing device). We consider the performance averaged across all videos of the database in Figure 6.3 (a) and across content category in Figure 6.3 (b). As a result, the optimal set evaluated by the proposed optimisation achieves a lower distortion with respect to benchmark solutions (especially compared to the Apple set). Most importantly, the optimal set achieves a substantial saving in terms of cost. While Netflix and Apple sets spend respectively around 5.4$ and 9$ to store a short ODV of 20 seconds in length, we ensure the same performance in terms of WS-PSNR while saving 50%-70% of their cost per sequence. This translates to a gain of 50-100 to store the entire dataset of 15 videos (*i.e.*, 300

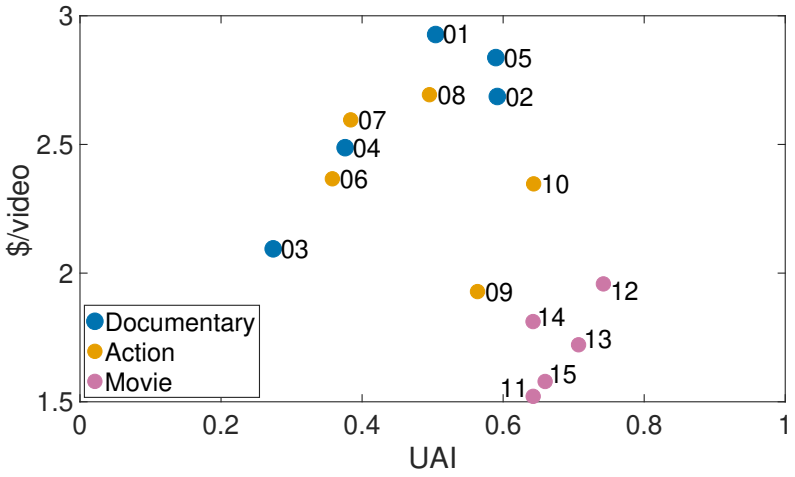
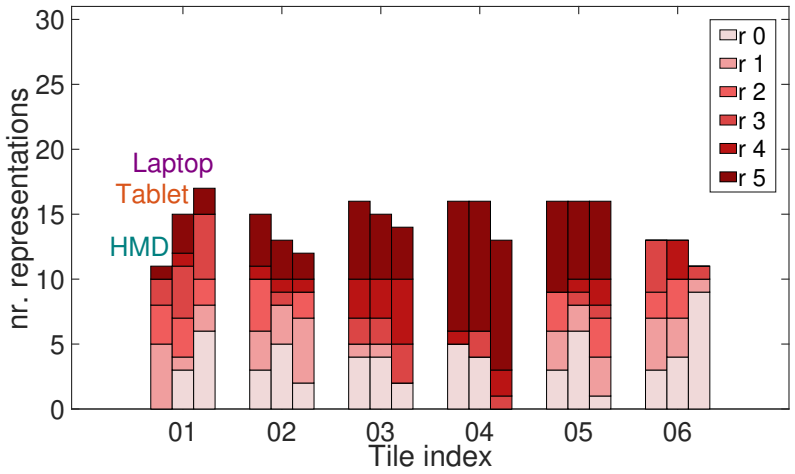


Figure 6.4: Total cost of storage and coding of optimal tile-representation set ($\lambda = 0.5$) per each video and User Affinity Index (UAI) averaged per devices.

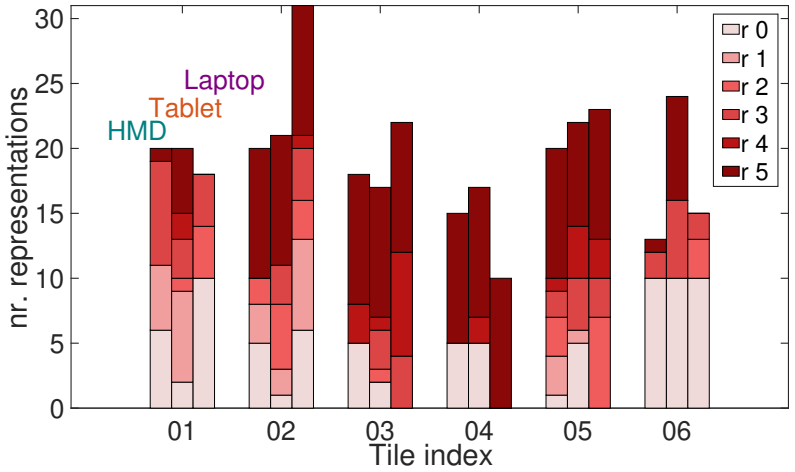
sec. of video content), which represents a significant saving in terms of cost even for the relatively small database presented in this work. If we imagine applying this optimisation to a bigger dataset and/or longer sequences, the financial saving could be very significant. The experienced quality is also strongly related to the video content as evident in Figure 6.3 (b). For instance, the *Movie* category is characterised by a reduced video complexity (see Chapter 4, Figure 4.6) and achieves higher performance with respect to the other video categories. More in general, for all video categories, the optimal set and the vendor recommendations achieve a comparable quality of experience, but with a much higher cost for the vendor ones. Finally, it is worth noting that when the representation set is optimised without taking into account user navigation, see black dot in Figure 6.3 (a), it performs almost as well as the optimal set with $\lambda = 0.5$ in terms of quality but it costs more than the double ($4.2 and $2, respectively). Overall, the optimised set of representations to store at the main server outperforms the recommended sets in terms of quality and, especially, total costs.

We are now interested in formalising the link between the data analysis provided in Chapter 4 (Section 4.6) and the user-centric server optimisation. For more in-depth study of the relationship between users behaviour and the final quality, Figure 6.4 depicts the total cost (per video) of the optimal tile-representation set optimised with $\lambda = 0.5$ as a function of the mean value UAI previously defined in Equation (4.3) of Chapter 4. As a remainder, the UAI represents the weighted average of cluster popularity (*i.e.*, how many users per cluster). In detail, the UAI approaches 1 when a small number of clusters with a large number of users per cluster are detected showing high affinity among users; on the contrary, UAI tends towards 0 when participants experience highly scattered navigation patterns, and they cannot

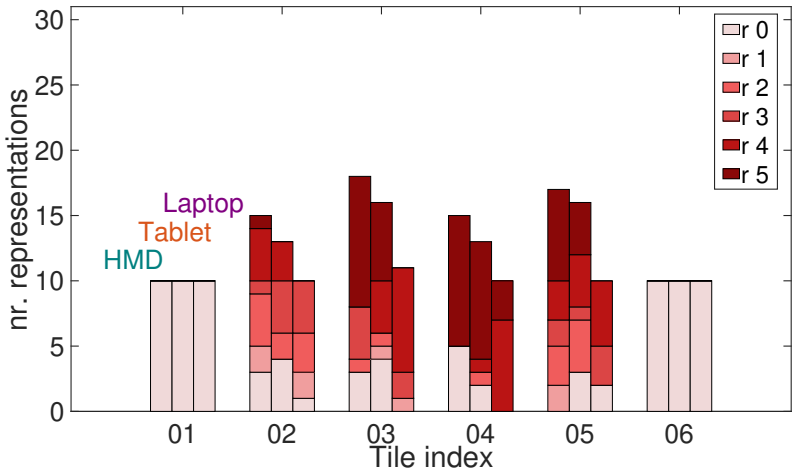
be clustered together. With the exception of IDs 09 and 10, in Figure 6.4 the total cost increases accordingly with the value of UAI, especially when observing per video category. This shows that the way in which users interact with the content influences the performance of the optimal set of tile representations stored in an adaptive ODV streaming system. We now investigate this intuition in-depth, providing an exhaustive analysis of the effect of users behaviour on the optimal set. In particular, we select three ODVs (namely, IDs 03, 08 and 13), each one coming from one different category. These videos are selected as heterogeneous samples −in terms of UAI and cost value ($) in Figure 6.4. The quality distribution over time and space of the optimal set evaluated with $\lambda = 0.5$ is now further analysed. As previously highlighted in the dataset behavioural analysis in Chapter 4 (Figure 4.7 (b)), users tend to display the central area (*i.e.*, around π value of latitude) of the equatorial zone in all ODVs of our database. This preference is reflected in the optimal tile set. Specifically, Figure 6.5 provides the stored coding rate level averaged over time per each tile and viewing device (*i.e.*, variable r in Table 6.1) computed by the proposed user-centric optimisation algorithm. At first look, it can be noticed that tiles corresponding to the two poles (*i.e.*, tile indexes 1 and 6) are mainly stored with the lowest value of quality. This is extremely evident for *Movie* sequences in Figure 6.5 (c). In contrast, the central area, such as tiles of index 3 and 4, have the majority of stored representations at the highest quality (*i.e.*, $r = 06$, where r is the selected coding rate as defined in Table 6.1). It is also worth mentioning that tile 4 (*i.e.*, one of the two frontal tiles as showed in Figure 6.2) is mainly selected either with the highest or lowest quality in all three examples. This could be related with the user probability of displaying that area. The algorithm allocates the highest quality to this tile since it is the most commonly selected one during the navigation, but to ensure the streaming service in all conditions, it also picks the lowest quality, which has the lowest cost. In the following, we further investigate this behaviour by observing the quality levels stored over time. For the sake of brevity and motivated by the previous observations of Figure 6.5, we now consider only HMD as viewing device, and we restrict the analysed area to the equatorial zone (*i.e.*, tile index 2, 3, 4 and 5). In Figure 6.6 (a,c,e), the UAI over time is compared with the total stored bitrate of the optimal tile set for Video IDs 03, 08 and 13, respectively. Interestingly, a strong correlation between these two metrics can be observed. For example, Video ID 13 of the *Movie* category has a high UAI, and the total stored bitrate is almost constant over time. In the other two examples, the amount of stored data is more sensitive to users' behaviour. A similar correlation can be noticed when comparing the UAI over time with the stored quality distribution in Figure 6.6 (b,d,f). In Figure 6.6 (b), we can note that diversity in



(a) Documentary (ID 3)



(b) Action (ID 8)



(c) Movie (ID 13)

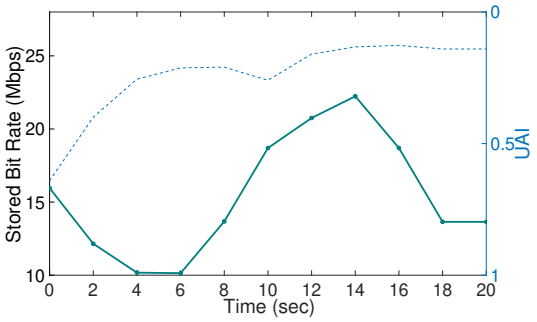
Figure 6.5: Total number of stored tile-representations for all rendering devices per a single video of each category. In particular, the three column represents the optimal set for each tile for all device corresponding to HMD, tablet and laptop in order from left to right.

terms of quality for the tile-representations is high when the affinity among users is low overall. In contrast, the Video ID 08 in Figure 6.6 (d) has a medium level of affinity but the variance of the stored quality levels is lower. Interestingly comparing Figure 6.6 (c) and (d), we can note that the UAI has a peak around 12–14 *sec.* leading to a drop in the stored bitrate (Figure 6.6 (c)). The behaviour may seem contradictory, but it is worth mentioning that a high affinity value means a reduced uncertainty in the system. Therefore, the resources can be better allocated based on users' preferences. Indeed, observing Figure 6.6 (d), the quality distributions of tile-representations is almost constant. Therefore, the lower value of stored bitrate around second 14 is due to a further reduction of stored representations in the polar area. As it is unlikely they will be selected by users, their quality level drops. The corresponding plots of Figure 6.6 evaluated for the other devices (*i.e.*, tablet and laptop) are provided in Figure 6.7 and 6.8, respectively; similar conclusions can be extracted from these last figures.

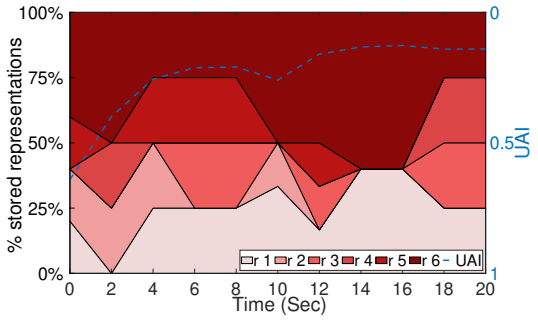
In summary, from this user-centric server optimisation, we can deduce the following:

- **Observation 1**: A significant saving in terms of bitrate and encoding/storage cost is achieved when the stored representations are optimised based on both content and users' profiles.
- **Observation 2**: The users behaviour generally affects the resource allocation of the optimal set (e.g., number of representations and quality levels).
- **Observation 3**: UAI provides a good representation of the existing correlation between users' behaviour and optimal set, floating the idea that UAI could be a key metric in the design of the next generation systems.

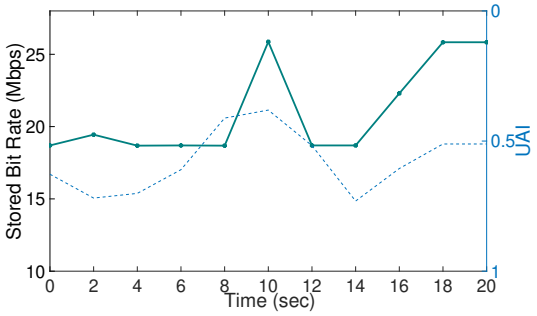
While Observation 1 has been already demonstrated in previous works examining conventional video [83] and ODVs [84], the other outcomes are novel insights that prove the importance of considering users behaviour in the design of a VR streaming system.



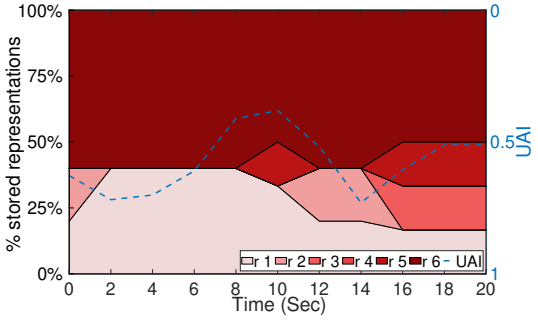
(a) Documentary (ID 03): total bitrate



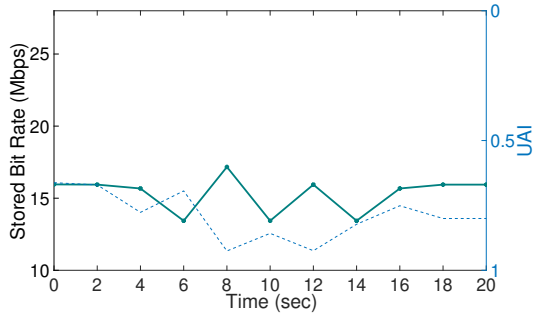
(b) Documentary (ID 03): bitrate level distribution



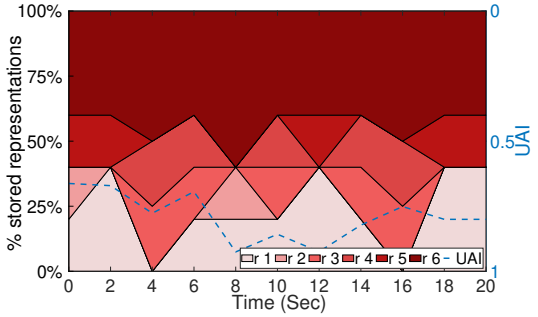
(c) Action (ID 08): total bitrate



(d) Action (ID 08): bitrate level distribution

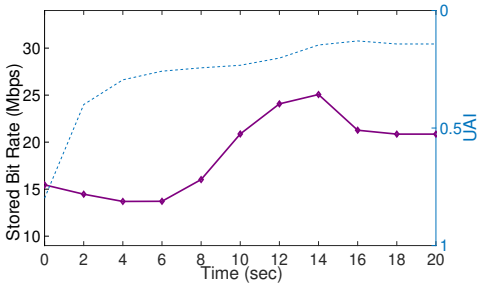


(e) Movie (ID 13): total bitrate

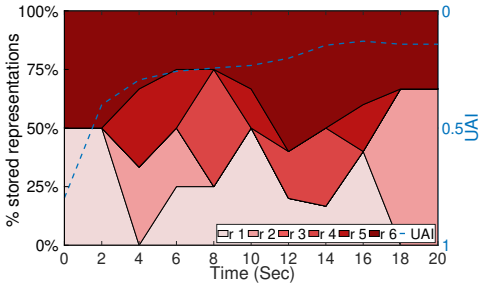


(f) Movie (ID 13): bitrate level distribution

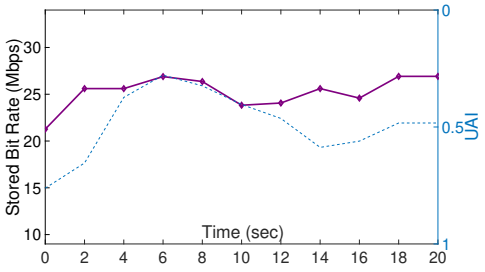
Figure 6.6: Temporal analysis of optimal tile-representation set for navigation trajectories with HMD across video categories. In the left column, the total stored bitrate over time for each video is presented while in the right column there is the bitrate level distribution of only equatorial area for each selected video. In each plot, the UAI over time is also reported.



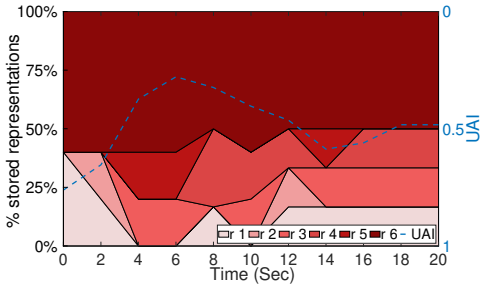
(a) Documentary (ID 03): total bitrate



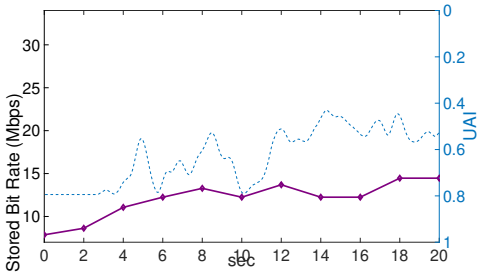
(b) Documentary (ID 03): bitrate level distribution



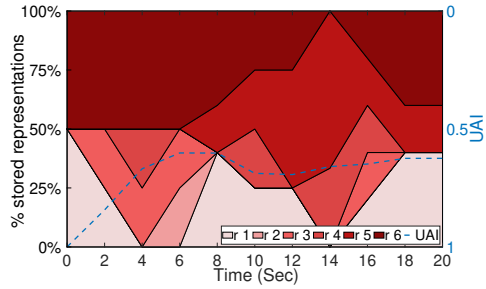
(c) Action (ID 08): total bitrate



(d) Action (ID 08): bitrate level distribution

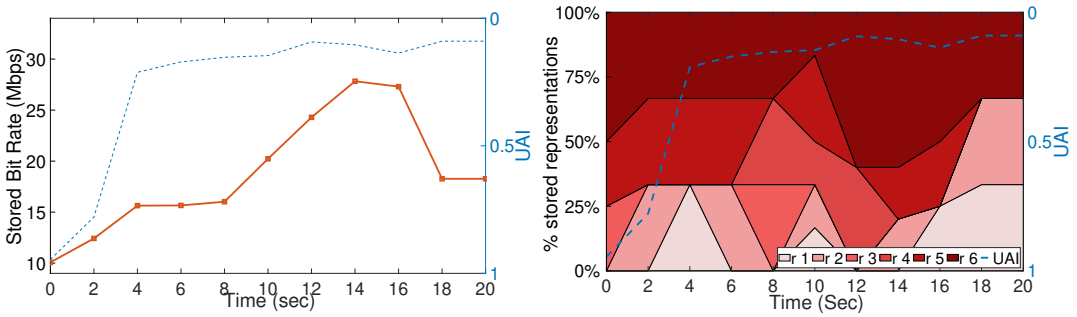


(e) Movie (ID 13): total bitrate



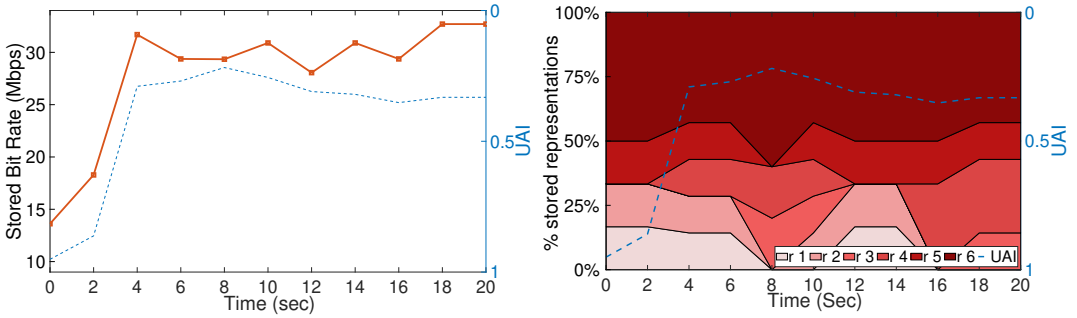
(f) Movie (ID 13): bitrate level distribution

Figure 6.7: Temporal analysis of optimal tile-representation set for navigation trajectories with Tablet across video categories. In the left column, the total stored bitrate over time for each video is presented while in the right column there is the bitrate level distribution of only equatorial area for each selected video. In each plots, UAI over time is also reported.



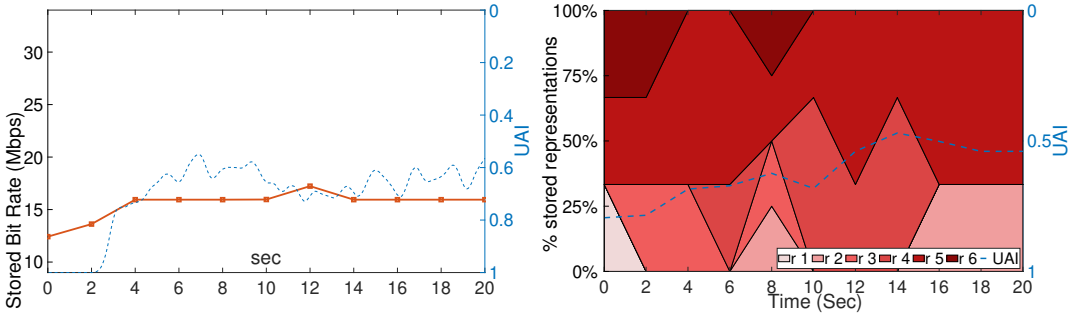
(a) Documentary (ID 03): total bitrate

(b) Documentary (ID 03): bitrate level distribution



(c) Action (ID 08): total bitrate

(d) Action (ID 08): bitrate level distribution



(e) Movie (ID 13): total bitrate

(f) Movie (ID 13): bitrate level distribution

Figure 6.8: Temporal analysis of optimal tile-representation set for navigation trajectories with Laptop across video categories. In the left column, the total stored bitrate over time for each video is presented while in the right column there is the bitrate level distribution of only equatorial area for each selected video. In each plots, UAI over time is also reported.

6.7 Chapter Summary

The overall goal of this chapter was to explore the impact of the way in which people navigate within ODV on the performance of 3-DoF VR adaptive streaming systems. Thus, we proposed a case study on the open problem of optimising the storage at the server provider for ODV adaptive streaming systems. A novel *user-centric immersive algorithm* has been proposed to optimise the set of VR representations to be stored at the server, minimising the total cost and yet maximising the final quality. The key-novelty of our algorithm is to take into consideration users behaviour beyond the spherical geometry and content information. As result, our optimal representation set ensures the same quality experienced with vendor recommendations but saving up to 70% of coding and storage cost. Leveraging on the novel dataset presented in Chapter 4, we have also shown how the different types of viewing devices (*e.g.*, HMD, laptop and tablet) but also user navigation (*e.g.*, affinity) impact on the optimal set. This opens the gate to a possibility of user-centric studies focused on making the users behaviour (and user affinity) the driver of VR system designs. To support this statement, we present in the next chapter a second case study of user-centric optimal strategy but this time from the client side (*i.e.*, adaptation logic) of a 3-DoF VR system.

Chapter 7

Navigation-Aware Adaptive Streaming Strategies

This chapter presents a second case study of user centric solution for 3-DoF VR systems. In particular, we propose an *optimal navigation-aware transmission strategy* able to fulfil the bandwidth requirements, while optimising the end-user quality experienced in the navigation.

7.1 Introduction

Nowadays, the multimedia format for VR applications is based on omnidirectional content, where a 360° scene is acquired instantaneously by an omnidirectional camera. The immersive sensation typical of VR is provided by placing the user at the center of the sphere and dynamically altering the portion of spherical content on display (*viewport*) according to the head direction of the user. Such a dynamic behaviour has posed novel questions on how to most efficiently utilise the available network resources. In particular, transmission of the entire panorama, even if only a small portion of it is actually displayed, guarantees zero latency for the user when switching viewing direction. However, this comes at the price of a poor quality, being the panorama sent at low quality for poor channel resources. A more efficient usage of bandwidth would be to exclusively send the viewport of interest. However, the viewport needs to be prefetched in advance, when the viewport requested by the user is not known yet but rather predicted. An erroneous prediction of the displayed viewport would require a re-transmission of a new predicted viewport, leading to large switching delays. Therefore, it is essential to seek the correct streaming strategy able to find the optimal trade-off between bandwidth efficiency, quality and latency, since the way in which users consume videos while navigating is highly dynamic and uncertain. In this work, we propose a novel transmission strategy able to directly address this trade-off in the case of HTTP adaptive streaming (HAS)

systems – *i.e.*, Dynamic Adaptive Streaming over HTTP (DASH) [40].

HTTP adaptive streaming systems offer users the possibility to adaptively select different versions (*i.e.*, different coding rates and resolutions) of video streams that have been pre-encoded and stored at the content distribution server. Based on the experienced channel, each media client optimises the appropriate version in order to maximise the video quality experienced, while navigating the scene. Along this direction, initial steps have been made with the study of adaptive streaming strategies for ODV [56, 82, 135, 201]. The work in [82] optimises DASH systems for omnidirectional content, but focuses mainly on the server side of the chain. Namely, the optimal storage strategy is investigated in the case that the panorama representations are encoded over unequal quality levels (*i.e.*, an area with high quality and the rest of the panorama at a low quality). A more formal tile-based DASH system is presented in [56, 201], where the new extension of DASH, defined as Spatial Relationship Description (SRD), is applied to the 360° video sequences. Both works focus on algorithms to generate tiles on the sphere, but crucially, no optimal strategy to select the optimal tile-representation at the client side is proposed. A tile-based adaptive streaming method is proposed in [135], where each user receives only the tiles that overlap with the predicted display viewport. This strategy, while effective from both a bandwidth and quality perspective, strongly depends on the viewport prediction.

This work proposes a *navigation-aware adaptation strategy* for 360° video adaptive streaming when sequence delivery is required for interactive users, aiming to provide a solution to the previously outlined challenge. In more details, we consider the scenario of 360° video sequences stored at the main server of the service provider (*e.g.*, Netflix, YouTube). Each acquired 360° video is projected onto a plane called *panorama* and is then processed by a tile-based encoder. Each tile is encoded at a different coding rate and resolution, creating per-tile representations. Each representation is then decomposed into temporal chunks (usually $2s$ long) and stored at the server. Based on his own future navigation path, the client requests the best set of per-tile representations for the entire panorama. From the panorama, the viewport of interest is then rendered and displayed. Figure 7.1 depicts the considered scenario in this work. The best set of representations downloaded by the user is defined as the one that (*i*) satisfies the channel bandwidth constraints and (*ii*) minimises the distortion of the most likely displayed viewports, while also reducing the distortion variations along most likely navigation paths. To achieve this goal, we evaluate the quality metric as a geometry based MSE to consider not only the content characteristics (*i.e.*, coding artifacts on the panorama) but also the scene geometry (*i.e.*, the projection of portions of the panorama on the sphere). Next, we

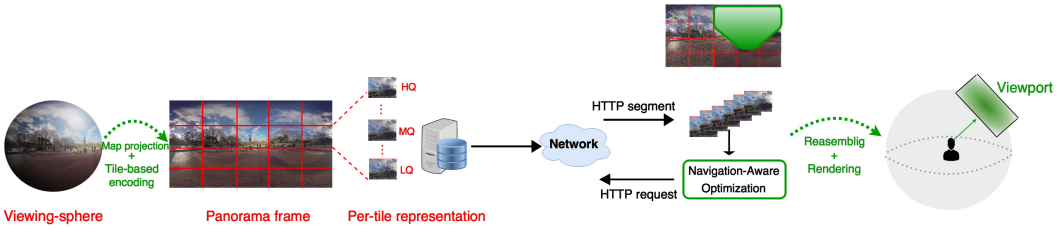


Figure 7.1: Overview of proposed architecture from viewing sphere to viewport display.

provide a formal problem formulation of the client adaptation logic and cast the problem as an ILP framework, which can be easily solved using the CPLEX solver. We further compare our adaptation logic strategy with the case of non-tile based coding. Simulation results show significant gains (in terms of navigation quality and smoothness) under different streaming scenarios. This reflects a more effective adaptation of the available network resources and furthermore, higher satisfaction experienced by the end-users. Finally, we compare the impact of different tile sizes on the final quality perceived by the user, showing that the optimal size depends on both the content characteristics as well as on the users interaction.

7.2 System model

We now provide an overview of the navigation-aware adaptive streaming system proposed in this work. We first describe the structure of adaptive streaming systems, and then outline the key features of the omnidirectional content.

7.2.1 Adaptive Streaming over HTTP

In adaptive streaming systems, one video sequence is divided into chunks of fixed duration (typically $2s$), and the number of chunks into which each sequence is decomposed is denoted by K. In the case of omnidirectional sequences, each chunk consists of T panoramic frames. Each panoramic frame F_t with $t = 1, ..., T$ is decomposed into N regular blocks (or tiles). Each tile is encoded into Q per-tile representations, with coding rates defined by the following set of rates $\mathcal{R} = \{R_1, R_2, ..., R_Q\}$[1]. Without loss of generality, we assume the system at regime (no rump-up or re-buffering phase) in which one chunk is periodically downloaded every chunk duration. Therefore, while displaying a chunk, the client downloads the following one, asking for the set of representation resulting from the adaptation logic optimization. The adaptation logic optimizes the representation vector $\mathbf{r} = [r_1, r_2, .., r_N]$, where $r_n \in \mathcal{R}$ represents the coding rate for the n^{th} tile of chunk to download. The optimization resulting in the best set of representations

[1]In this paper, we do not vary the encoded resolution across representations. However, our optimization problem can be directly extended to provide a solution in a scenario considering multiple resolutions.

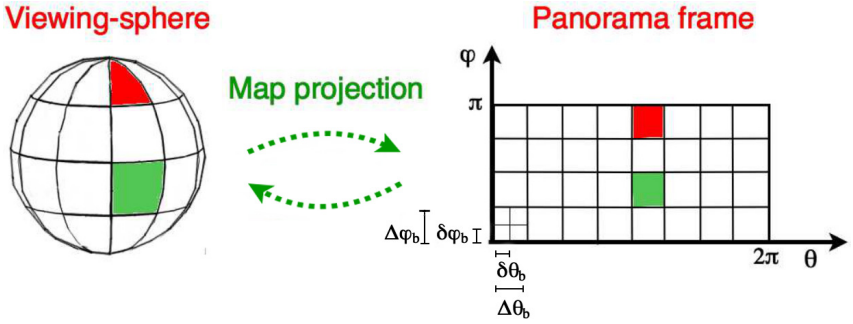


Figure 7.2: Map projection from the viewing sphere to the panorama image.

to download for each chunk is what we propose in this paper. At each downloading opportunity, the user knows the popularity of each viewport to be displayed (*heatmap*) as well as the rate-distortion function for each tile-representation for the chunk of interest. This information can be periodically delivered to clients through the media presentation description, and it can reflect the information for each chunk or be averaged over a set of chunks (i.e., a trade-off between communication overhead and optimization accuracy). Equipped with this information, the optimization proposed in the following sections is invoked and the optimal chunk is requested for downloading.

7.2.2 Omnidirectional Video

We consider an acquired spherical video projected into rectangular panoramic frames (*map projection*) via an equirectangular projection[2], since it is the simplest and most popular map projection [202]. In particular, a point on the viewing sphere can be mapped onto the panorama through longitude ($0 \leq \theta \leq 2\pi$) and latitude ($0 \leq \phi \leq \pi$) values. Then, each panoramic frame is processed by a tile-based encoder with uniform tiles. Therefore, panoramic frames are decomposed into blocks of area $S_b = \Delta\theta_b \Delta\phi_b$ where $\Delta\theta_b$ and $\Delta\phi_b$ are the longitudinal and latitudinal dimensions of each block. Because of the map projection, these blocks on the viewing sphere have unequal sizes. Each block can be seen as the sum of infinitesimal elements with dimension $\delta\theta_b$ and $\delta\phi_b$. Therefore each block centred in (θ_b, ϕ_b) has an area on the sphere given by $\mathcal{S}_b = l^2 B^2 sin\phi_b \delta\theta_b \delta\phi_b$, where l is the radius of the sphere and B the number of infinitesimal elements per block[3]. In particular, the blocks near the poles are smaller than those in the equatorial zone, as can be inferred from Figure 7.2.

At the client side, any viewers, equipped with a head mounted device, nav-

[2]Note that the optimisation problem proposed in this work is general enough to be extended to any other map projection method.

[3]We denote $\mathcal{S}$ for any given surface value on the sphere, while S represents any surface value on the panoramic image.

igates the 360° video by moving their head and changing the displayed viewport accordingly. The viewport is a plane tangent at the viewing sphere in the user's view direction (θ_i, ϕ_i), as shown in Figure 7.1. In particular, its longitudinal and vertical resolutions are imposed by the user's screen and denoted by $\Delta\theta_v$ and $\Delta\phi_v$, respectively. Considering the sphere with unitary ray ($l = 1$), we denote by $\mathcal{VP}_i$ the viewport with centre in (θ_i, ϕ_i) with $i = 1, ..., I$,[4] and its surface on the sphere is equal to:

$$\mathcal{S}_{\mathcal{V}_i} = \int_{\theta_i - \frac{\Delta\theta_v}{2}}^{\theta_i + \frac{\Delta\theta_v}{2}} \int_{\phi_i - \frac{\Delta\phi_v}{2}}^{\phi_i + \frac{\Delta\phi_v}{2}} \sin\phi d\theta d\phi \,. \tag{7.1}$$

Each viewport consists of a set of blocks (or tiles). Therefore, let us denote by $\mathcal{S}_{b_n}$, a surface on the sphere of the n^{th} block centered in (θ_n, ϕ_n), given by:

$$\mathcal{S}_{b_n} = \int_{\theta_n - \frac{\Delta\theta_b}{2}}^{\theta_n + \frac{\Delta\theta_b}{2}} \int_{\phi_n - \frac{\Delta\phi_b}{2}}^{\phi_n + \frac{\Delta\phi_b}{2}} \sin\phi d\theta d\phi \tag{7.2}$$

and $\alpha_{n,i}$ is the portion on the sphere of the n^{th} block overlapping with the viewport $\mathcal{VP}_i$.

7.3 Geometry-based QoE metric

We now define two quality metrics that describes the objective function in our optimisation: (*i*) the *popularity-weighted geometry-based distortion*, *i.e.*, the distortion of the different regions of the sphere associated to each possible viewport, weighted by the probability that the user selects that specific viewport, and (*ii*) the *navigation-smoothness*, *i.e.*, the variation of the geometry-based distortion experienced during the navigation.

7.3.1 Popularity-weighted geometry-based distortion

Firstly, we define the distortion experienced by the user while navigating in the 360° video and we highlight the differences with respect to the distortion of the decoded panoramic frame. In term of notation, in the following we adopt $\mathcal{D}$ to indicate any distortion values on the sphere and D to indicate the distortion on the panoramic image.

We assume that the distortion of a given viewport is measured by the distortion of the portion of the sphere which underpins the viewport. Therefore, the distortion

[4]We denote the total number of directions that we sample on the sphere I. Ideally, $I \to \infty$, but in practice the head position is quantized.

of a generic viewport $\mathcal{VP}_i$ with its center in (θ_i, ϕ_i) is evaluated as:

$$\mathcal{D}_i = \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \int_{\theta_i - \frac{\Delta\theta_v}{2}}^{\theta_i + \frac{\Delta\theta_v}{2}} \int_{\phi_i - \frac{\Delta\phi_v}{2}}^{\phi_i + \frac{\Delta\phi_v}{2}} \mathcal{D}(\theta, \phi) \sin\phi d\theta d\phi \tag{7.3}$$

where $\mathcal{D}(\theta, \phi)$ is the distortion function at any point (θ, ϕ) on the viewing sphere. Decomposing the viewport into the different blocks derived from the tile-based coding, Equation (7.3) can be reformulated as:

$$\mathcal{D}_i = \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n \in \mathcal{VP}_i} \int_{\tilde{\theta}_{i,n}^-}^{\tilde{\theta}_{i,n}^+} \int_{\tilde{\phi}_{i,n}^-}^{\tilde{\phi}_{i,n}^+} \mathcal{D}(\theta, \phi) \sin\phi d\theta d\phi \tag{7.4}$$

where $\tilde{\theta}_{i,n}^- = \min(\theta_i - \frac{\Delta\theta_v}{2}, \theta_n - \frac{\Delta\theta_n}{2})$, $\tilde{\theta}_{i,n}^+ = \min(\theta_i + \frac{\Delta\theta_v}{2}, \theta_n + \frac{\Delta\theta_n}{2})$, and similarly $\tilde{\phi}_{i,n}^- = \min(\theta_i - \frac{\Delta\theta_v}{2}, \theta_n - \frac{\Delta\theta_n}{2})$, $\tilde{\phi}_{i,n}^+ = \min(\theta_i + \frac{\Delta\theta_v}{2}, \theta_n + \frac{\Delta\theta_n}{2})$. Recalling that $\alpha_{n,i}$ is the percentage of block n that overlaps with the portion of the sphere underpinning viewport $\mathcal{VP}_i$, the previous equation can be further generalised as follows:

$$\mathcal{D}_i = \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n=1}^{N} \alpha_{n,i} \int_{\theta_n - \frac{\Delta\theta_v}{2}}^{\theta_n + \frac{\Delta\theta_v}{2}} \int_{\phi_n - \frac{\Delta\phi_v}{2}}^{\phi_n + \frac{\Delta\phi_v}{2}} \mathcal{D}(\theta, \phi) \sin\phi d\theta d\phi \tag{7.5}$$

where the summation has been extended to all blocks within the panorama. All pixels on the sphere in the range $\{[\theta_n - \frac{\Delta\theta_v}{2}, \theta_n + \frac{\Delta\theta_v}{2}], [\phi_n - \frac{\Delta\phi_v}{2}, \phi_n + \frac{\Delta\phi_v}{2}]\}$ belong to the block n on the panorama, which has been encoded at the same rate for each representation level. The representation encoded at rate r_n will lead to a distortion averaged over the block denoted by $D_n(r_n)$. From this consideration as well as from Equation (7.2), the distortion of viewport $\mathcal{VP}_i$ is given by

$$\begin{aligned}
\mathcal{D}_i(\mathbf{r}) &= \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n=1}^{N} D_n(r_n) \alpha_{n,i} \int_{\theta_n - \frac{\Delta\theta_v}{2}}^{\theta_n + \frac{\Delta\theta_v}{2}} \int_{\phi_n - \frac{\Delta\phi_v}{2}}^{\phi_n + \frac{\Delta\phi_v}{2}} \sin\phi d\theta d\phi \\
&= \frac{1}{\mathcal{S}_{\mathcal{V}_i}} \sum_{n=1}^{N} D_n(r_n) \alpha_{n,i} \mathcal{S}_{b_n} \\
&= \sum_{n=1}^{N} D_n(r_n) \alpha_{n,i} \widehat{\mathcal{S}}_{n,i}
\end{aligned} \tag{7.6}$$

where $\widehat{\mathcal{S}}_{n,i} = \mathcal{S}_{b_n} / \mathcal{S}_{\mathcal{V}_i}$ is the block surface on the sphere normalized by the area of the viewport, and where we explicitly show the dependency of $\mathcal{D}_i$ on $\mathbf{r}$.

The probability for the user to display viewport $\mathcal{VP}_i$ in the panoramic frame t is denoted by $p_{t,i}$, and hence, the popularity-weighted distortion of the chunk to be

downloaded is:

$$\mathcal{D}(\mathbf{r}) = \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{n=1}^{N} D_n(r_n) \widehat{\mathcal{S}}_{n,i} \alpha_{n,i} p_{t,i}. \tag{7.7}$$

It is worth noting that the rate-distortion on the panorama block $D_n(r_n)$ does not depend on the time index t since $D_n(r_n)$ reflects the mean distortion of block n encoded at the coding rate r_n for all frames in the chunk.

7.3.2 Navigation-smoothness

Beyond the average quality experienced during the navigation, we are interested in evaluating the quality variation, since variation of quality while changing viewport can result in an annoying degradation in quality of experience.

Given a user who is displaying $\mathcal{VP}_i$ at time t, we evaluate the distortion variation between two consecutive viewports displayed at time $t-1$ and t. This is given by:

$$\Delta\mathcal{D}_{t,i}(\mathbf{r}) = \sum_{j \in \mathcal{N}(i)} |\mathcal{D}_i(\mathbf{r}) - \mathcal{D}_j(\mathbf{r})| \, p_{t-1,j}$$

where $\mathcal{N}(i)$ is the set of viewports that could have been displayed at time $t-1$ and is defined as the set of viewports with center in (θ_j, ϕ_j) such that

$$\begin{cases} \theta_j \leq \theta_i \pm \theta_{head} \\ \phi_j \leq \phi_i \pm \phi_{head} \end{cases} \tag{7.8}$$

where θ_{head} and ϕ_{head} are the maximum angular movements of the human head between two consecutive frames. The navigation-smoothness per chunk can then be evaluated as follows:

$$\begin{aligned} \Delta\mathcal{D}(\mathbf{r}) &= \sum_{t=2}^{T} \sum_{i=1}^{I} \Delta\mathcal{D}_{t,i}(\mathbf{r}) p_{t,i} \\ &= \sum_{t} \sum_{i} \sum_{j \in \mathcal{N}(i)} \left| \sum_{n} D_n(r_n) (\widehat{\mathcal{S}}_{n,i} \alpha_{n,i} - \widehat{\mathcal{S}}_{n,j} \alpha_{n,j}) \right| p_{t-1,j} p_{t,i} \end{aligned} \tag{7.9}$$

7.4 Navigation-Bandwidth Adaptive Logic

Equipped with the above metrics and notations, we can now formulate the optimisation problem that needs to be solved at the client side at each downloading opportunity. In the following, we first formalise the optimisation problem and we then describe the solving method.

7.4.1 Problem formulation

We seek the optimal set of representations for all blocks of the panoramic frames such that the quality experienced in the scene navigation is maximised and yet the bandwidth constraint is respected. We can then express the navigation-aware adaptation logic optimisation for each chunk as:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \mathcal{D}_{user}(\mathbf{r}) \\ \text{s.t.} \quad & \sum_{n} r_n \leq C \end{aligned} \tag{7.10}$$

where C is the estimated channel capacity during the delivery of the chunk of interest and $\mathcal{D}_{user}(\mathbf{r})$ is the metric that takes into account both the geometry-based quality and the navigation-smoothness. In particular,

$$\begin{aligned} \mathcal{D}_{user}(\mathbf{r}) &= \mathcal{D}(\mathbf{r}) + \lambda \Delta \mathcal{D}(\mathbf{r}) \\ &= \sum_{t} \sum_{i} \big[\mathcal{D}_i(\mathbf{r}) + \lambda \Delta \mathcal{D}_{t,i}(\mathbf{r})\big] p_{t,i} \end{aligned} \tag{7.11}$$

where λ is the multiplier that allows us to assign an appropriate weight to the quality in the objective metric. Parametrizing the rate-distortion function of the panorama blocks leads to the following [174]:

$$D_n(r_n) = a_n + \frac{b_n}{r_n + c_n} \tag{7.12}$$

and hence, the problem formulation in (7.10) becomes:

$$\begin{aligned} \min_{\mathbf{r}} \quad & \mathcal{D}_{user}(\mathbf{r}) \\ s.t. \quad & \sum_{n} \frac{b_n}{D_n(r_n) - a_n} - c_n \leq C \end{aligned} \tag{7.13}$$

where a_n, b_n and c_n are constants that depend on the content characteristics of block n.

The above optimization problem is computationally complex to solve being $\mathcal{D}_{user}$ neither a convex nor a linear function. In the following, we show how to cast the problem in (7.13) in a tractable ILP optimization problem.

7.4.2 ILP Optimization Algorithm

We recall that the set of representations available for each block is finite and corresponds to a specific set of coding rates $\mathcal{R}$ used to store the representations at the server. It follows that, in the panoramic frame, the distortion of each block $D_n(r_n)$

can be expressed as:

$$D_n(r_n) = \sum_{q=1}^{Q} D_n(R_q)\beta_{n,q} \tag{7.14}$$

where $R_q \in \mathcal{R}$, and $\beta_{n,q} = 1$ if $r_n = R_q$, $\beta_{n,q} = 0$ otherwise. This means that rather than seeking the best coding rate $\{r_n\}_n$ for all blocks in the panorama, we seek the best set of binary variables $\{\beta_{n,q}\}_{n,q}$. Adopting a change of variable $x_{n,q} \rightarrow D_n(R_q)$, the objective function becomes:

$$\mathcal{D}_{user}(\mathbf{r}) = \sum_{t=1}^{T}\sum_{i=1}^{I}\left[\sum_{n=1}^{N}\sum_{q=1}^{Q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} + \sum_{j\in\mathcal{N}(i)}\left|\sum_{n=1}^{N}\sum_{q=1}^{Q} x_{n,q}\beta_{n,q}(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j})\right| p_{t-1,j}\right] p_{t,i}$$

The previous expression is not linear because of the absolute value in the second term. However, an equivalent objective function linear in $\beta_{n,q}$ can be evaluated as shown in the following. We introduce an auxiliary variable y such that:

$$y = \sum_{n=1}^{N}\sum_{q=1}^{Q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}}_{n,i}(\alpha_{n,i} - \alpha_{n,j}) \tag{7.15}$$

The absolute value in (7.15) can then be obtained by imposing the two following constraints on the y variable:

$$y_{i,j} \geq \sum_{n}\sum_{q} x_{n,q}\beta_{n,q}(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j})$$
$$y_{i,j} \geq -\left(\sum_{n}\sum_{q} x_{n,q}\beta_{n,q}(\widehat{\mathcal{S}}_{n,i}\alpha_{n,i} - \widehat{\mathcal{S}}_{n,j}\alpha_{n,j})\right) .$$

Finally, the optimisation problem in (7.10) can be casted as an ILP problem shown in (7.16) (see Table 7.1). The objective function minimises the expected quality experienced by the user when navigating the scene in the chunk duration. The constraint (7.16a) guarantees that only one representation is selected for each block in a chunk, while (7.16b) imposes the bandwidth constraint. Finally, the constraints (7.16c) and (7.16d) are the terms of transformation of the absolute value in a linear function.

Table 7.1: ILP problem formulation for a navigation-aware adaptive logic.

Integer Linear Programming

$$\min_{\beta,\mathbf{y}} \sum_{t}\sum_{i}\left[\sum_{n}\sum_{q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}_n}\alpha_{n,i} + \sum_{j\in\mathcal{N}(i)} y_{i,j}p_{t-1,j}\right]p_{t,i} \tag{7.16}$$

$$\text{s.t.} \sum_{q}\beta_{n,q} = 1, \qquad \forall n \in [1,N] \tag{7.16a}$$

$$\sum_{n}\sum_{q}\left(\frac{b_n}{x_{n,q}-a_n} - c_n\right)\beta_{n,q} \leq C \qquad \forall n \in [1,N] \tag{7.16b}$$

$$y_{i,j} \geq \sum_{n}\sum_{q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}_n}(\alpha_{n,i}-\alpha_{n,j}) \tag{7.16c}$$

$$\forall t \in [1,T], \forall i \in [1,I], \forall j \in \mathcal{N}(i), \forall n \in [1,N]$$

$$y_{i,j} \geq -\left(\sum_{n}\sum_{q} x_{n,q}\beta_{n,q}\widehat{\mathcal{S}_n}(\alpha_{n,i}-\alpha_{n,j})\right) \tag{7.16d}$$

$$\forall t \in [1,T], \forall i \in [1,I], \forall j \in \mathcal{N}(i), \forall n \in [1,N]$$

7.5 Simulation Results

7.5.1 Simulation Setups

We consider two 360° videos, namely *"Rollercoaster"* and *"Timelapse NY"*. Both the sequences have been downloaded in equirectangular format at the maximum spatial resolution and frame rate available on the platform YouTube, *i.e.*, 3840x2048 pixels and 30 fps, respectively. The sequences have been selected because of their different spatial and temporal complexity. In particular, *"Rollercoaster"* is more complex since it has a moving camera and its values of SI and TI are equal to 72 and 45, respectively. On the contrary, *"Timelapse NY"* has a fixed camera that shoots city streets and its corresponding SI and TI values are 44 and 14, respectively.

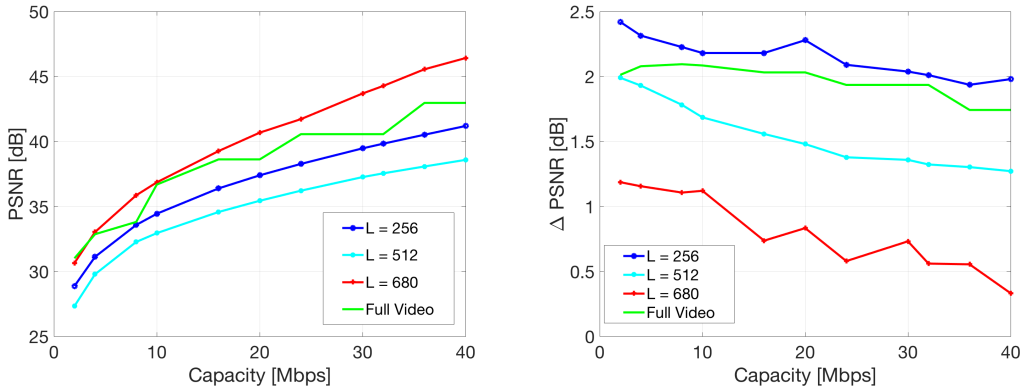
To simulate a tile-based encoding, sequences have been split temporally and spatially in blocks. This results in a reduced coding efficiency with respect to a standard tile-based encoder. Therefore, the gain provided in following should be considered as lower bound to the actual gains, which can further improve in the case of more efficient tile-based coding strategies. We set a chunk of duration of about $2s$ and squared blocks with three different sizes, $L = [256, 512, 680]$ pixels. We then compare our optimised strategy with a baseline case in which the entire

panorama is encoded (without tile-based encoding) at the same average rate. We label this baseline method by "Full Video" in the following results. Each block (as well as the entire panorama) has been encoded with HEVC codec [203] with an overall coding rate ranging between 16 *kbps* and 150 *Mbps*. We then consider 15 representations for each blocks ($Q = 15$). These representations are selected as the one corresponding to quality levels (in terms of Peak Signal-to-Noise Ratio (PSNR)) of $[25, 28, 29, 30, 31, 32, 33, 34, 35, 38, 40, 42, 45, 50, 52]$ dB. The rate value associated to each quality score has been derived by the rate-distortion function given in Equation (7.12), where the parametric values are evaluated by curve fitting.

As input of our ILP problem, the prediction of user's navigation path in the 360° content is required. Using the free software Graph-Based Visual Saliency (GBVS) [204], we computed for each panoramic frame the position of each FoA. From this FoA map, we derived the heatmap over time. The two considered videos differ substantially in terms of resulting heatmap over time. The *"Rollercoaster"* sequence has one main FoA, which leads to a nicely predictable behaviour of the users. On the contrary, *"Timelapse NY"* has several FoAs, increasing therefore the uncertainty of the interactivity behaviour of the users. Finally, the selection of the most suitable set of representations-per-block is optimised with our ILP optimisation problem in scenarios characterised by values of C ranging from 2 *Mbps* to 40 *Mbps*. Moreover, we assign a unitary weight to quality in the objective function of our problem ($\lambda = 1$). We have used the generic solver IBM ILOG CPLEX [193] to solve the ILP proposed in this work. Results in the following are provided both for the quality (in terms of PSNR) and for the navigation-smoothness (in terms of PSNR difference) and they have been carried out by over 100 simulated interactive users downloading over a constant channel constraint over time. It is worth noting that our simulation considers some approximations (infinite playback buffers, exact channel estimation, etc.) with respect to real HAS systems. But these do not impact on our objective in this paper, which is to demonstrate the benefit of considering content and interactive information in the optimal representation selection for a HAS client in a stationary regime.

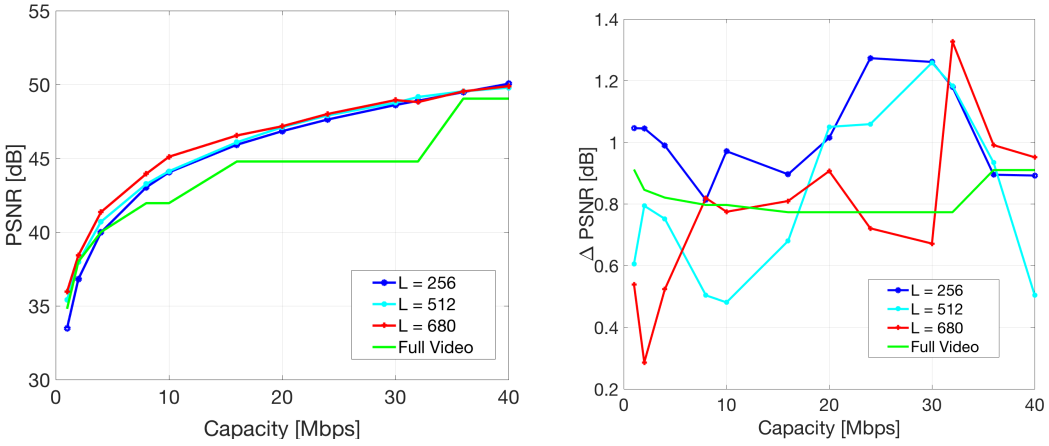
7.5.2 Results

In Figure 7.3, both the quality (in terms of PSNR) defined in Equation (7.6) and the navigation-smoothness (in terms of PSNR difference) have been provided as a function of the available bandwidth, for the *"Rollercoaster"* video sequence. As expected, the quality increases with the available bandwidth, Figure 7.3 (a). Most importantly, the proposed optimisation with tile size $L = 680$ outperforms the "Full



(a) Viewport Quality **(b)** Viewport Quality Variation

Figure 7.3: Analysis of Rollercoaster with $\lambda = 1$ and 100 users.



(a) Viewport Quality **(b)** Viewport Quality Variation

Figure 7.4: Analysis of Timelapse NY with $\lambda = 1$ and 100 users.

Video" case (with no tiling). This shows the gain of the added degree of freedom in the adaptation logic thanks to the tiling. However, by decreasing the tile size, this quality gain fades away. This is motivated by the fact that tiling leads to a more flexible transmission strategy, but at the price of a reduced coding efficiency. This tradeoff is overall good for $L = 680$ and not for $L = 512$ and $L = 256$. In particular, in this type of sequences in which the FoA is very narrow and uniform across users, there is no need of too much refined tiles (*i.e.*, small values of L). Therefore, the loss in coding efficiency due to small L value is not necessarily balanced by the gain in the adaptation logic. A similar trend is observed for the smoothness-navigation, where $L = 680$ reduces the quality variations experienced during the navigation of the 100 randomly generated users. A slightly different behaviour is observed in the case of the *"Timelapse NY"* sequence, Figure 7.4. For

values of capacity bigger than 5 *Mbps*, each tiled solution achieves a better final quality than in the delivery of the entire encoded panorama. This is due to (i) different video characteristics that lead to a different penalty in coding efficiency, (ii) different navigation patterns of the interactive users. The distribution of FoA is far more variable then the case of *"Rollercoaster"* and misses a dominant area of interest. Therefore, higher resolution in optimizing the per-tile representation (small L values) balance the loss in coding efficiency and lead to a quality gain with respect to the no-tile case (Full Video). However, the quality variations observed in Figure 7.4b are highly random. This can be mainly justified by the fact that in the case of multiple FoAs predicting the users navigation path only from the heatmap (as we assume in our problem formulation) is not enough reliable. This shows the need for an improved prediction model to be adopted in our representation optimisation.

7.6 Chapter Summary

In this chapter, we have presented a novel navigation-aware strategy for 360° video adaptive streaming. In particular, we have proposed an adaptation logic at the client side able to choose the best set of representations-per-block to download, in order to achieve an optimal final quality. We have evaluated the performance of our algorithm comparing the final quality of different tile sizes with the entire encoded video. Even if a visible gain in terms of navigation quality is provided, the results shows also to be strongly affected by the content of sequences and user navigation. Therefore, this theoretical framework has highlighted the benefit of taking into account of the behavioural preferences of interactive users into the adaptation logic at the client side, confirming the need of developing user-centric solutions for 3-DoF VR system.

Part IV

TOWARDS BEHAVIOURAL ANALYSIS IN 6-DoF VR

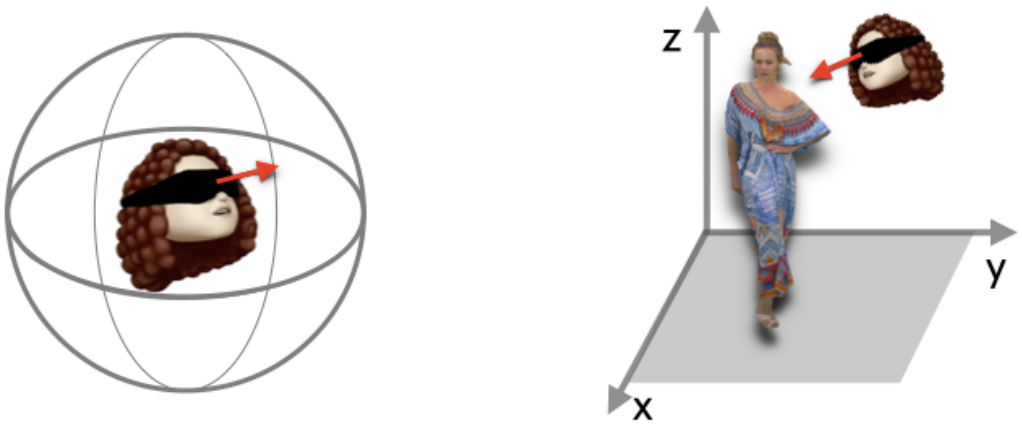
Chapter 8

From 3-DoF to 6-DoF: new metrics to analyse immersive users

This last part of the thesis is aimed at enabling user behavioural analysis of VR trajectories while displaying dynamic volumetric media in 6-DoF conditions. Specifically, we first extend the applicability of existing behavioural methodologies adopted for studying user behaviour in 3-DoF settings to 6-DoF scenario (Chapter 8). Then, we present a specific case study of behavioural analysis in a social VR movie where people were enable to navigate with 6-DoF (Chapter 9).

8.1 Introduction

Immersive reality technology has revolutionised how users engage and interact with media content, going beyond the passive paradigm of traditional video technology, and offering higher degrees of presence and interaction in a virtual environment. Depending on the enabled locomotion functionalities in the 3D space, immersive environments can be classified as 3- or 6-DoF. In the first scenario, the de-facto multimedia content is the *omnidirectional*. The viewer is fully immersed in a virtual space where they can navigate and interact thanks to an immersive device – typically an HMD, which enables to display only a portion of the environment around him/herself, named *viewport*. As shown in Figure 8.1 (a), the media is displayed from an *inward* position, and the viewer can interact with the content only by changing the viewing direction (*i.e.*, by looking up/down or left/right or tilting the head side to side). In a 6-DoF system, the user can also change viewing perspective by moving (e.g., walking, jumping) inside the virtual space. The scene is therefore populated by *volumetric objects* (*i.e.*, meshes or point clouds) which are observed from an *outward* position (Figure 8.1 (b)). This extra degree of freedom brings the virtual experience even closer to reality: a higher level of interactivity makes the user feels more immersed and present within the virtual environment [205].



(a) 3-DoF **(b)** 6-DoF

Figure 8.1: Viewing paradigm in 3- and 6-DoF VR.

Despite their differences, the common denominator of both interactive systems is the viewer as an active decision-maker of the displayed content. This active role of the user defines the *user-centric* era, in which content preparation, streaming, and rendering need to be tailored to the viewer interaction to remain bandwidth-tolerant whilst meeting quality and latency criteria. Media codecs need to be optimised in such a way that the quality experienced by the user is maximised [71, 106]. Analogously, to ensure high-quality content and smooth navigation, but remaining bandwidth-tolerant [30, 90, 92], streaming should be tailored to users interactivity. The latter however is highly dependent on the user navigation within the content which is not known a priori. Here is an urgent need to understand, analyse and predict users behaviour [34, 95, 96].

Thanks to the large availability of public datasets [109, 114, 116, 119], user navigation in 3-DoF immersive systems has been deeply investigated, showing the importance of analysing and detecting key behavioural aspects in interactive (user-centric) systems, as described in Chapter 3. However, the 6-DoF counterpart is not yet considered in the literature apart from some few cases [206–208]. User navigation in 6-DoF scenarios was also studied in the past in the context of locomotion and display technology for CAVE environments [209, 210]. However, the focus has been mainly put on the analysis of completion time per task versus different setting conditions. While highly informative to summarise the interaction of users within a content, these metrics usually fail in providing other key information: which users navigate similarly within the content, and which are the dominant interaction behaviours among users. The importance of this information has been already proved in 3-DoF, and deeply investigated in this thesis with a spherical clustering algorithm and an information-theoretic approach proposed in Chapter 4 and 5 of this thesis, respectively. Thus, this behavioural investigation has been instead overlooked in the

emerging 6-DoF environment.

In this chapter, we want to fill the gap of behavioural analysis in 6-DoF system. The main research question we aim to address is how new physical settings and locomotion functionalities given to users can affect the analysis and understanding of their behaviour and how these limitation can be overcame to enable such behavioural analysis. Therefore, we also focus on extending the applicability of clustering methods to investigate users similarity (*i.e.*, users sharing common behaviours while interacting with the content) in a 6-DoF environment. Specifically, clustering techniques usually rely on pairwise similarity metrics, and at the moment there is no a proposed metric to measure the interactivity similarity between two users in 6-DoF. Starting from state-of-the-art clustering that we developed for 3-DoF (Chapter 4), we describe the main limitations of the tool when extended to 6-DoF, and we propose a new methodology for overcoming those limitations. In detail, we explore how different distances features (*i.e.*, user positions in the 3D space, user viewing directions) but also distance measurements (*i.e.*, Euclidean, Geodesic distance) can be used to model consistent viewport overlap. Using a publicly available dataset of navigation trajectories in 6-DoF [92], we study how spherical clustering solutions fare when applied to the 6-DoF setting. Results indicate that 3-DoF clustering solutions are not able to capture similarities when users are placed at far distances between each other, suggesting that new solutions tailored for 6-DoF navigation are needed. Thus, we define the exact user similarity metric, which we will be considering as our ground truth. Given its computational complexity, we propose a simpler and yet reliable proxy for it. More concretely, we define and compare 8 different *similarity metrics* which are based on different *distance features* and *distance measurements*. We validate and test our proposed similarity metrics on a publicly available dataset of navigation trajectories collected in a 6-DoF Virtual Reality (VR) scenario [92]. Results have shown that similarity metrics based on different distance features are promising solutions to correctly detecting users with a similar behaviour while experiencing volumetric content. Finally, we validate the proposed tool by testing it on navigation trajectories collected in a different setting, a 6-DoF Augmented Reality (AR) scenario [208]. Similarities among users are detected as well in this new interactive setting, showing that the proposed metric is general to be efficient in multiple interactive systems with 6-DoF.

In conclusion, our work contributes to the overall open problem of behavioural analysis in a 6-DoF system, with the following main contributions:

- formal definition of user trajectory and ground-truth user similarity (in terms of overlap of the display content) in 6-DoF, formally highlighting the main

difference between 6-DoF and 3-DoF;

- a deep-in comparison on how new physical settings and locomotion functionalities given to users can affect the analysis;
- an exhaustive analysis of different metrics capturing users trajectory similarity (in terms of distance on the plane or from the object) and the ability to approximate the ground truth. This analysis based on 6-DoF VR trajectories reveals the only position on the floor is not sufficient to characterise the user behaviour and the viewing direction cannot be neglected;
- a case study of behavioural analysis in an AR system via a state-of-the-art clustering tool using our proposed similarity metrics.

The remainder of this chapter is organised as follows: related works on user behavioural analysis in both 3-DoF and 6-DoF systems are reported in Section 8.2. The main challenges of detecting behavioural similarities in a 6-DoF system and the importance of having a tool that approximates such similarities are described in Section 8.3. A first analysis of the relationship between viewport overlap and distance features and measurements is given in Section 8.4. Then, we describe our proposed and validated similarity metrics on real navigation trajectories collected in a 6-DoF VR settings in Section 8.5 and Section 8.6, respectively. In Section 8.8, a case of study is presented to show the applicability of our proposed metrics also to an 6-DoF AR setting. Our results are further discussed in Section 8.9. Conclusions of this chapter are summarised in Section 8.10.

8.2 Related Work

8.2.1 User Behaviour in 3-DoF environment

The user navigation within a 3-DoF environment has been intensely analysed from many perspectives. Many studies have been focused on psychological investigations of user engagement and presence correlated to movements within the spherical content. In [211], a study from a large scale experiment (511 users and 80 omnidirectional videos) showed the positive correlation between lower interactivity level and higher engagement level (strong focus on few points of interest). Similarly, a correlation between the perceived sense of presence and the interactivity level was detected in [212], with more random exploratory interactions for less immersed (and hence less engaged) users. However, none objective metric to properly quantify and characterise the user behaviour has been presented in these works. To further understand how people observe and explore 360° contents, many public

datasets of navigation trajectories have been made available. Those datasets usually come with statistical analysis aimed at capturing average users behaviour, as a function of maximum and average angular speeds under various video segment lengths [109] or eye fixation distribution [116]. A deeper analysis was presented in [119] where the dataset has been analysed through a clustering algorithm presented in Chapter 4, specifically built to have in the same cluster users who similarly explore 360° content. The analysis validated previous understanding that movies with few focus of attention lead to higher engagement, in this case, shown by users sharing strong similarities and hence collected into few and high-populated clusters. However, behavioural analysis based on such clustering tool mainly provides a general idea of similarity among viewers without offering however a quantitative metric. To overcome such limitation, we showed in Chapter 5 the benefit of studying spatio-temporal trajectories by information theory metrics, and thus the possibility of identifying and quantifying behavioural aspects. Key outcomes from this quantitative analysis were the study of similarities between users when watching the same content, but also the similarity of a given user when watching diverse content. The importance of these behavioural insights has been then exploited in different VR applications. For instance, authors in [140] proposed a scalable prediction algorithm for user navigation, which considered previous navigation patterns while in [213] an hybrid approach has been presented based on both dominant user behaviour (detected via a clustering approach) and the video content. Moreover, the analysis and understanding of user navigation in a VR environment has shown promising results also in determining the mental health issues of subjects (*e.g.*, anxiety, eating disorders, depression) and their treatment [214, 215].

8.2.2 User Behaviour in 6-DoF environment

Extending such behavioural analysis to a 6-DoF environment is not straightforward, due to the change in the viewing paradigm (from inward to outward) and to the addition of translation in 3D space. In the past, user navigation in 6-DoF scenarios was studied in the context of locomotion and display technology for CAVE environments [209, 216]. A Cave Automatic Virtual Environment (CAVE) system is an immersive room on which walls and floor are projected the video content and viewers are free to move inside [217]. For instance, the study is [209] focused on the task performance analysis in terms of completion time and correct actions. Authors in [216] compared instead the effect of two different immersive platforms such as CAVE and HMD on the user navigation. More traditional metrics, such as angular distance and linear velocity, alongside completion time, were also used to compare different navigation controllers (*i.e.*, joystick-based vs head-controlled navigation)

in 6-DoF [218]. In detail, authors showed the superiority of head-controlled techniques, allowing more sense of presence and better control with less discomfort in the navigation. While the aforementioned analysis tools are highly informative to summarise the interaction of users within a 6-DoF environment, they usually fail in providing other key insights: which users navigate similarly, and which are the dominant interaction behaviour among users.

Recently, the focus has been put on subjective quality assessment based on different coding techniques of volumetric content, both static [206] and dynamic [92, 207]. These studies present a preliminary statistical analysis of user movements in terms of mean angular velocity, most displayed areas of the content showing an influence in the navigation due to the perceived content quality, and a preference to visualise the volumetric object from a close and frontal perspective. This last finding was also confirmed in a behavioural navigation analysis while consuming volumetric video content by an AR mobile application [208]. Here, viewers movements were analysed in terms of distribution on the floor, viewing angles, and relative distance from the content. All these preliminary studies are based on traditional metrics for behavioural analysis, which consider only one user feature at the time, either position on the floor or viewing direction but not together, suffering from the major shortcomings highlighted before. In this chapter, we aim to overcome these limitations by proposing a generalisable and efficient tool for detecting similar viewers while experiencing 6-DoF content.

8.3 Challenges

In this work, our main goal is to define a new pairwise metric able to capture the (dis)similarity between two 6-DoF users (in terms of displayed content). This metric needs to be reliable and yet simple to compute.

In the following, we first define our assumption of similarity among users while navigating in a 6-DoF environment. Then, we propose an exact user similarity metric highlighting its limitations, and therefore the need to find a simpler and reliable proxy for it. Finally, we show the advantages of having a similarity metric for behavioural analysis via a clique-based clustering approach presented in Chapter 4, which identified users who are attending the same portion of an omnidirectional content in a 3-DoF system. This clustering technique relies on a pairwise similarity metric, and thus, having a proper metric also for 6-DoF system would extend the applicability of this state-of-the-art tool.

8.3.1 User Similarity in 6-DoF

We are interested in analysing user behaviour, assuming that users interact similarly when they *observe the same volumetric content.* The user behaviour can be identified by the spatio-temporal sequences of their movements within the virtual environment, namely *navigation trajectories*. For simplicity, we consider only one object of interest in an otherwise empty 3D scene of a 6-DoF system. Our analysis can be straightforwardly extended to multiple objects in the same scene.

In a 3-DoF scenario, the trajectory of a generic user i can be formally denoted by the sequence of the user's viewing direction over time $\{p_1^i, p_2^i, .., p_n^i\}$ where p_t^i is the centre of the viewport projected on the immersive content (*i.e.*, spherical video) at timestamp t. The point p can be represented in spherical coordinates by $[\theta, \phi, r]$ where $\theta \in [0, 2\pi]$ is the azimuth angle (or longitude), $\phi \in [0, \pi]$ the polar angle (or latitude), and r is the distance between the point (viewport center projected on the immersive content) and the origin (user position). In a 3-DoF scenario, users are positioned at the centre of the spherical content; thus, r is constant during the interaction. As a consequence, the viewport centre alone is highly informative of the user behaviour and can be used as a proxy of viewport overlap among users, as we shown in Chapter 4. In particular, the geodesic distance has been proved as a reliable similarity metric such that low value indicates high similarity between 3-DoF users.

In a 6-DoF setting, the distance between the user and immersive content can change over time due to the added degrees of freedom. Thus, the more degrees of freedom are given to the user, the more challenging becomes the system and the description of user navigation within it. The viewport centre alone is no more sufficient to characterise the user behaviour in a 6-DoF scenario. Figure 8.2 shows an example of two users navigating in a 6-DoF system. In the bottom part of the figure, there are navigation trajectories of two users i and j projected on a 2-D domain (*i.e.*, floor). Each point x_t represents the spatial coordinates (*i.e.*, [x,y,z]) on the floor of viewers while each associated vector symbolises the viewing direction. In the top part of Figure 8.2, we have instead a snapshot of a specific time instant t. In more detail, the shaded triangular areas represent the *viewing frustum* per user, which indicates the region within the user viewport, and r_t is the distance between the user and the volumetric content. We have also depicted the viewport centre p_t projected on the displayed volumetric object. Given the two users i and j at time t, in the case of $r_t^i \gg r_t^j$, the user j (very close to the object) is visualising a very focused and detailed part of it; conversely, user i is pointing to the same area but from a much further distance, thus she/he is experiencing the content differently with less defined details. Despite this difference, the small distance $D_t(i, j)$ between viewport cen-

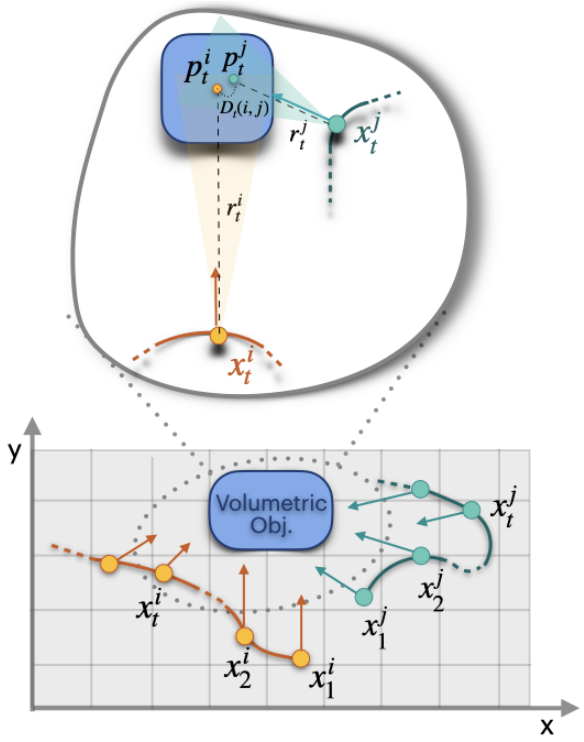


Figure 8.2: An example of 6-DoF trajectories projected in a 2D domain for user i and j. In the circle, a snapshot at time t where coloured triangles represent viewing frustum per user.

tres p_t^i and p_t^j might suggest a high similarity (*i.e.*, high viewport overlap) between the corresponding users, which does not reflect the reality in the case of $r_t^i \gg r_t^j$. Thus in this scenario, we cannot rely on the viewport centre only to characterise the user behaviour. The distance r and the spatial coordinates on the virtual floor x are also needed. Given the above notation, we can formally define the navigation trajectory for a generic 6-DoF user i as $\{(x_1^i, p_1^i, r_1^i), (x_2^i, p_2^i, r_2^i), \ldots, (x_n^i, p_n^i, r_n^i)\}$. This information is crucial to define a simple similarity metric among users in this new setting.

8.3.2 Overlap Ratio as the ground-truth metric

Since we are interested in capturing viewers that are attending similar volumetric content at the same time instance, the straightforward measure that could show this behaviour is the overlap among viewports. Given two users i and j described above in Figure 8.2 (top part), we denote their displayed viewport as $\mathcal{S}_t^i$ and $\mathcal{S}_t^j$, respectively, defined as the set of points of the volumetric content falling within their viewing frustum. Then, we denote the overlap set by $\mathcal{S}_t^i \cap \mathcal{S}_t^j$, defined as the portion of points displayed by both users. Equipped with the above notation, we can now introduce a key metric for the analysis: the *overlap ratio* $O(i,j)$. This is defined as the cardinality of the overlap set, normalised by the cardinality of the

set containing all points of the volumetric content visualised by both users. More formally, the overlap ratio in a specific time t is:

$$O_t(i,j) = \frac{|\mathcal{S}_t^i \cap \mathcal{S}_t^j|}{|\mathcal{S}_t^i \cup \mathcal{S}_t^i|} \tag{8.1}$$

where $\mathcal{S}_t^i$ and $\mathcal{S}_t^j$ are the displayed viewport of users i and j, respectively. The higher is the overlap ratio, the higher is the similarity between users, and vice versa. Even if this metric is exact and a clear indicator of how much similar users are with respect to their displayed content, its evaluation is not trivial as it is intensely time-consuming. For instance, the overlap ratio between two users requires 0.8986 seconds per frame on average on an Intel R machine with CPU E5-4620 at 2.10 GHz; the operation needs to be computed for all the possible combinations of users, leading to a large overhead which does not meet requirements for real-time and scalable applications. A new measure is thus needed to perform clustering in near real-time.

8.3.3 Clustering as a tool for behavioural analysis

Being able to assess users similarities in an objective way might be crucial for different applications such as behavioural analysis. As defined in Chapter 4, a clique-based clustering algorithm is used to detect users with similar behaviour. This requires a reliable graph to be constructed in such a way that only the nodes that identify similar users (*i.e.*, who are displaying the same portion of the content) are connected. Equipped with such a meaningful graph, the clique-based clustering identifies optimal sub-graphs of all inter-connected nodes, ensuring the identification of the largest cluster of users all sharing a large viewport overlap. In more detail, given a set of users who are experiencing the same content, we can represent their movements in a time-window T as a set of graphs $\{\mathcal{G}_t\}_{t=1}^{T}$. Each unweighted and undirected graph $\mathcal{G}_t = \{\mathcal{V}, \mathcal{E}_t, \mathrm{A}_t\}$ represents behavioural similarities among users at time t, where $\mathcal{V}$ and $\mathcal{E}_t$ denote the node and edge sets of $\mathcal{G}_t$, respectively. Each node in $\mathcal{V}$ corresponds to a user interacting with the content. Each edge in $\mathcal{E}_t$ connects neighbouring nodes defined by the binary adjacency matrix A_t. Assuming that users are connected if they are displaying similar content, we can formally define the adjacency matrix A_t as follow:

$$\mathrm{A}_t(i,j) = \begin{cases} 1, & \text{if } w_t(i,j) \leq G_{th} \\ 0, & \text{otherwise}\,. \end{cases} \tag{8.2}$$

(a) *Longdress* (PC1) **(b)** *Loot* (PC2) **(c)** *Red and Black* (PC3) **(d)** *Soldier* (PC4)

Figure 8.3: Human Body Point Clouds content used in the analysed public available dataset.

where $w_t(i,j)$ is a similarity metric between user i and j and G_{th} is a thresholding value. On this final graph, the clique-based clustering algorithm can be applied to identify a set of users all connected (*i.e.*, clique), and therefore with similar behaviour. In Chapter 4, we based this graph construction on a pairwise similarity metric specifically for the 3-DoF trajectories.

Identifying a generic and reliable metric $w(i,j)$ that approximates behavioural similarities among users who experience a 6-DoF content is a key step to enable user behavioural analysis via tools proposed for 3-DoF scenario and the focus on the next section.

8.4 A first attempt of behavioural analysis in 6-DoF

We now investigate how new settings and locomotion functionalities of 6-DoF users affect their behavioural analysis. Specifically, we present the twofold lines of behavioural investigations: the first aimed at identifying the most relevant distance metrics in a 6-DoF scenario; the second at proving that a tool for the behavioural analysis of 3-DoF trajectories need to be adjusted before to be used for 6-DoF trajectories. Table 8.1 defines the distance features and measurements that we consider in this chapter.

Table 8.1: Definition of distance features and measurements.

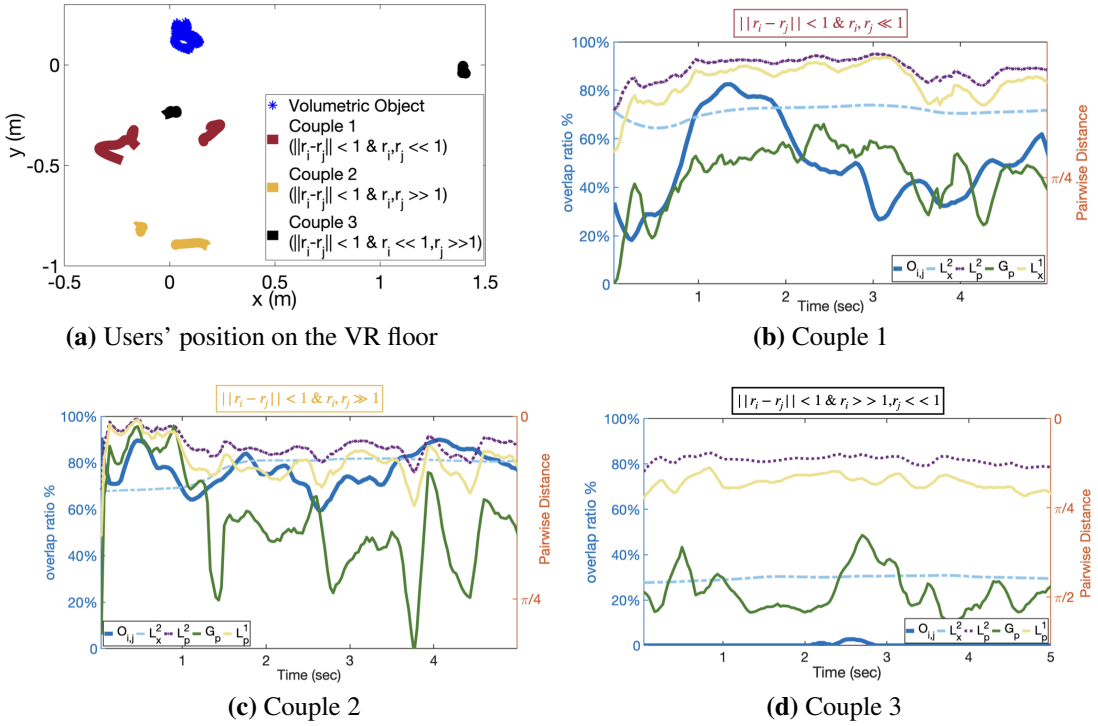
Symbol	Definition
x	user position on the VR floor
p	viewport center projected on the volumetric content
r	relative distance between user and volumetric content
$\mathsf{L}(\cdot,\cdot)$	difference of relative distance between two users
$\mathsf{E}(\cdot,\cdot)$	Euclidean distance
$\mathsf{G}(\cdot,\cdot)$	Geodesic distance

8.4.1 Dataset and Methodology

Dataset. Existing datasets with user navigation collected while displaying volumetric objects in a 6-DoF environment are still very limited. In the following, we use the open dataset presented in [92]. The dataset is comprised of navigation trajectories of 26 users participating in a visual quality assessment study in VR. For the study, four dynamic point cloud sequences were employed [219], namely *Long dress* (PC1), *Loot* (PC2), *Red and black* (PC3), *Soldier* (PC4) (Figure 8.3). Each sequence was distorted at four different bit rate points with two compression algorithms: the anchor used for the MPEG call for proposals, and the upcoming MPEG standard V-PCC. Hidden references were additionally employed in the test, for a total of 36 stimuli. Similarly to what is shown in Figure 8.2, a single object of interest was placed in the VR scene, and users were instructed to focus on the volumetric content for the duration of the session and rate its visual quality. Therefore, the navigation data adheres to the assumptions listed in Section 8.3.

Simple set of metrics. We assume that two generic users i and j of the dataset are placed at given time t in positions x_t^i and x_t^j, respectively. To verify if their overlap ratio defined in the previous section can be substituted with the distance between the two viewport centers $D(p_t^i, p_t^j)$, we consider 4 different metrics to take into account the heterogeneous shape of the PCs: the euclidean distance between users' position in the space (L_x^2), and the distance between the viewport centres projected on the volumetric content in terms of euclidean (L_p^2), geodesic (G_p), and cityblock distance (L_p^1). In particular, geodesic distance is the shortest arc length connecting the points on a sphere, while cityblock evaluates the absolute differences between coordinates.

Graph Construction. To implement the graph-based clustering proposed in Chapter 4, we need to construct a binary graph following Equation (8.2), as described in Section 8.3. In short, users with a similarity metric below a threshold value G_{th} are neighbours in the graph. Hence, the first step is to identify G_{th}. We empirically evaluate the ROC curves per each metric and select the best value (Fig-



(a) Users' position on the VR floor

(b) Couple 1

(c) Couple 2

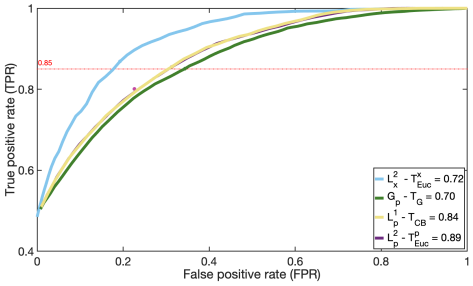
(d) Couple 3

Figure 8.4: Comparison between significant couples of users navigating in PC3 (*Red and black*).

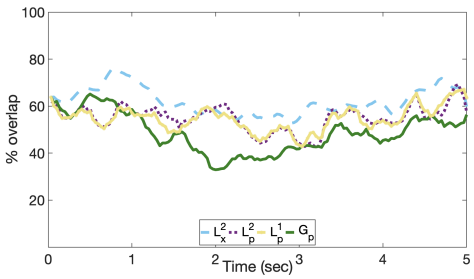
ure 8.5 (a)). We assess these values based on navigation trajectories collected for the entire dataset above described. As ground-truth for the ROC, we assumed that two users are attending the same portion of content if their viewports overlap by at least 75% of their total viewed area. The predicted event is instead evaluated using the eight metrics presented in the previous section that approximate the overlap. We selected threshold values to have a probability to correctly classify an event (*i.e.*, TPR) equal to 0.75.

8.4.2 Distance as a proxy for overlap?

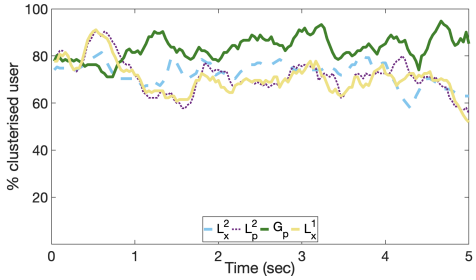
We conducted a first analysis of the relationship between viewport overlap and distance between the user and the volumetric object, by studying three couples of users with different behaviour. This difference lies mainly in the user position. In more details, we considered the following pair of users: **Couple 1:** users i and j sharing a similar position at a small distance from the object ($||r_i - r_j|| < 1 \; ; r_i, r_j \ll 1$); **Couple 2:** users i and j sharing a similar position at a large distance from the object ($||r_i - r_j|| < 1 \; ; r_i, r_j \gg 1$); **Couple 3:** user i (j) close to (far from) the object ($||r_i - r_j|| > 1 \;\; ; r_i \ll 1, \; r_j \gg 1$). Figure 8.4 (a) depicts the spatial position over time of the selected users' couples (given by their HMD position) with respect to the centroid of the volumetric content in the sequence PC3. Figure 8.4 (b-d)



(a) ROC curves per distance metrics. In the legend, threshold values for the spherical clustering.



(b) Mean overlap over time (Clusters > 2 users)



(c) Mean clustered user over time (Clusters > 2 users)

Figure 8.5: Spherical clustering results over time per sequence PC3 (*Red and black*).

compare the viewport overlap over time (expressed in percentage) for each couple ($O_{i,j}$, blue solid line), which represents our ground truth information, versus their distance $D(i,j)$ for the four different distance metrics described in the previous subsection. When users share a similar position (Figure 8.4 (b-c)), the correlation between pairwise overlap and distance metrics is quite evident (high overlap, low distance), especially when geodesic distance is considered and users are close to the object. Conversely, the euclidean distance between users is not so informative since is almost flat. In the context of the third couple, the overlap is negligible (given the quite different positions of the users from the object), but the distance metrics fail in capturing this behaviour. Finally, L^2_p and L^1_p work similarly in all cases, even though the overlap is substantially different (high in subfigure (b) and (c), very low in (d)). Only the geodesic distance between two viewport centres seems to be much higher compared with the previous couples.

8.4.3 Distance to assess users' similarity?

After showing that the previous metrics does not perfectly replicate the overlap behaviour, we now show why this is a fundamental problem when studying user behaviour. We do so by looking at user similarities via clustering techniques as defined in Section 8.3.

Using the threshold values computed as described in Section 8.4.1, we applied

Table 8.2: Spherical clustering analysis over time per each video content. The different distance metrics used as similarity matrices are considered.

(a) Results for the first two sequences (PC1 and PC2).

	PC1				PC2			
	L_x^2	L_p^2	L_p^1	G_p	L_x^2	L_p^2	L_p^1	G_p
Mean N. Tot Clusters	9.63	8.7	8.7	**6.2**	10.9	7.14	7.14	**6.76**
Mean N. Single Cluster (cl. = 1 user)	3.85	3.73	3.73	**1.90**	5.03	2.87	2.97	**2.20**
Mean Overlap within Cl. (cl. >2 user)	**62.84%**	59.73 %	59.59 %	49.31 %	**57.00 %**	40.19 %	40.01 %	42.05 %
Mean Clustered Population (cl. >2 user)	73.60 %	72.49 %	72.99 %	**85.95 %**	66.27 %	78.44 %	78.40 %	**78.96 %**

(b) Results for the last two sequences (PC3 and PC4)

	PC 3				PC 4			
	L_x^2	L_p^2	L_p^1	G_p	L_x^2	L_p^2	L_p^1	G_p
Mean N. Tot Clusters	10.05	8.81	8.85	**6.49**	10.91	9.61	9.54	**7.19**
Mean N. Single Cluster (cl. = 1 user)	4.18	3.61	3.60	**1.92**	4.77	3.97	3.98	**2.23**
Mean Overlap within Cl. (cl. >2 user)	**62.00 %**	55.04 %	54.62 %	48.48 %	**61.41 %**	54.51 %	55.19 %	46.95 %
Mean Clustered Population (cl. >2 user)	72.67 %	71.41 %	70.72 %	**83.41 %**	67.90 %	71.83 %	72.72 %	**84.22 %**

the spherical clustering at each content frame. To avoid misleading results with clusters composed of a single user, we only consider clusters composed of more than 2 users. At each frame, we evaluated the viewport overlap among all users within the same cluster and averaged across clusters. Figure 8.5 (b) shows this mean as a function of the time frame for the four distance metrics under consideration. In Figure 8.5 (c), instead, we measure how large clusters are on average. We depict this by plotting the percentage of users falling within each cluster (averaged over all clusters) as a function of time. We plot this for each of the four distance metrics considered. We observe that all metrics reach an average of viewport overlap within clusters between $40\% - 60\%$. Even if clusters based on L_x^2 seem to reach a higher overlap ratio within the same cluster, it is also relevant to notice that part of the user population is not covered, since they fall in small clusters (with less than 2 users). The percent of users took into account is indeed around 70% of the entire population (Figure 8.5 (c)). On the contrary, clustering based on the geodesic distance between viewport centres (L_p^2) finds larger clusters but less meaningful ones as it leads to a smaller mean overlap ratio. A global view of the results is offered in Table 8.2, which provides results (averaged over time) for all the sequences in the dataset. Results in the table confirm the previously observed trend: clusters based on the geodesic distance between viewport centres (G_p) are able to identify consistent groups of users, while those based on the euclidean distance between users (L_x^2) perform better in terms of viewport overlap. Here, the first limitation of the metrics currently available to analyse users behaviour in 6-DoF: the lack of one metric that can provide highly populated clusters (as we would like to identify mainstream interactivity) with a large overlap ratio between users within clusters (as we need to identify representative clusters). Equally important, despite the

Table 8.3: Similarity metrics: definitions, included distance features and measurements, regulator and threshold values.

Symbol	Definition	Distance Feature and Metric	Regulator values	G_{th}
w_1	$k_\alpha^{(\mathsf{E})}(x^i, x^j)$	$\mathsf{E}(x^i, x^j)$	$\alpha = 1$	0.64
w_2	$k_\alpha^{(\mathsf{L})}(r^i, r^j)$	$\mathsf{L}(r^i, r^j)$	$\alpha = 1$	0.80
w_3	$k_\alpha^{(\mathsf{G})}(p^i, p^j)$	$\mathsf{G}(p^i, p^j)$	$\alpha = 1$	0.63
w_4	$k_\alpha^{(\mathsf{E})}(p^i, p^j)$	$\mathsf{E}(p^i, p^j)$	$\alpha = 1$	0.84
w_5	$k_\alpha^{(\mathsf{E})}(x^i, x^j) \cdot k_\beta^{(\mathsf{L})}(r^i, r^j) \cdot k_\gamma^{(\mathsf{G})}(p^i, p^j)$	$\mathsf{E}(x^i, x^j)$, $\mathsf{L}(r^i, r^j)$, $\mathsf{G}(p^i, p^j)$	$\alpha = 0.1$; $\beta = 0.5$; $\gamma = 1$	0.54
w_6	$k_\alpha^{(\mathsf{E})}(x^i, x^j) \cdot k_\beta^{(\mathsf{L})}(r^i, r^j) \cdot k_\gamma^{(\mathsf{E})}(p^i, p^j)$	$\mathsf{E}(x^i, x^j)$, $\mathsf{L}(r^i, r^j)$, $\mathsf{E}(p^i, p^j)$	$\alpha = 0.1$; $\beta = 0.125$; $\gamma = 0.2$	0.87
w_7	$k_\alpha^{(\mathsf{E})}(x^i, x^j) \cdot \beta[\eta(r_i) + \eta(r_j)] \cdot k_\gamma^{(\mathsf{G})}(p^i, p^j)$	$\mathsf{E}(x^i, x^j)$, r^i,r^j, $\mathsf{G}(p^i, p^j)$	$\alpha = 0.25$; $\beta = 0.5$; $\gamma = 0.5$	0.60
w_8	$k_\alpha^{(\mathsf{E})}(x^i, x^j) \cdot \beta[\eta(r_i) + \eta(r_j)] \cdot k_\gamma^{(\mathsf{E})}(p^i, p^j)$	$\mathsf{E}(x^i, x^j)$, r^i, r^j, $\mathsf{E}(p^i, p^j)$	$\alpha = 0.5$; $\beta = 0.5$; $\gamma = 0.5$	0.62

metric used, the values of overlap ratio are below 63%. However, we recall that we set a distance threshold value corresponding to 80% overlap. Here the second limitation: current distance metrics are not a reliable proxy for the viewport overlap measure. As a consequence, this paper opens the door to a very new challenge on designing a proper metric to analyse users' behaviour in 6-DoF. The intuition is that this metric will need to consider both user positions (*i.e.*, x_i, x_j) and viewing directions (*i.e.*, p_i, p_j) to efficiently analyse 6-DoF users.

The core of this first attempt of behavioural analysis highlights the key differences in the interactivity models between 3-DoF and 6-DoF, showing that current metrics fail in capturing similarity among users (in terms of overlap of the displayed content), and thus existing clustering methodologies used in 3-DoF cannot be reliably extended to 6-DoF due to the lack of proper metrics. As consequence, we overcome these limitations by proposing new similarity metrics in order to verify which one better approximate similarity among user to enable a proper analysis of user behaviour in 6-DoF.

8.5 Proposed similarity metrics

In this section, we present eight similarity metrics and we provide an exhaustive study to understand which one approximates at the best the viewport overlap. Those metrics are expressed as a function of various *distance features* and *measurements* considering either users position on the floor (x) or users viewing direction in terms of viewport centre projected on the volumetric content (p) or both. Thus, we divide the metrics in two groups: *single-feature* and *multi-feature* metrics. Table 8.3 summarises our proposed similarity metrics.

8.5.1 Single-feature metrics to assess users similarity

The first set of similarity metrics are based on one single feature (*i.e.*, the distance of either viewport centres on the volumetric content or users on the floor). For the sake of notation, in the following, we omit the temporal parameter t. We model the

similarity functions via radial basis function kernel, and specifically the Gaussian kernel [220] defined as follows:

$$k_{\alpha}^{(D)}(i,j) = e^{-\alpha D(i,j)} \tag{8.3}$$

where $D(i,j)$ is the distance between two generic users i and j, while $\alpha > 0$ is a parameter to better regularise the distance. This distance can be evaluated in multiple ways. The first two similarity metrics w_1 and w_2 are based on the location of users in the virtual space with respect to the virtual object or other viewers. The former is based on the Euclidean distance $\mathsf{E}(x^i, x^j)$ between user i and j on the virtual floor, while w_2 considers the difference of users relative distance to the centroid of the displayed content, $\mathsf{L} = ||r^i - r^j||$. Specifically, we define them as follows:

$$w_1 = e^{-\alpha \mathsf{E}(x^i,x^j)} = k_{\alpha}^{(\mathsf{E})}(x^i, x^j)\,; \tag{8.4}$$

$$w_2 = e^{-\alpha ||r^i - r^j||} = k_{\alpha}^{(\mathsf{L})}(r^i, r^j)\,. \tag{8.5}$$

The metrics w_3 and w_4 are instead based on the distance between the two viewport centres of user i and user j projected on the volumetric content. To take into account the heterogeneous shape of the volumetric content, this distance in w_3 is evaluated in terms of the Geodesic distance $\mathsf{G}(p^i, p^j)$ while in w_4 in terms of the Euclidean distance $\mathsf{E}(p^i, p^j)$. More formally, they are defined as:

$$w_3 = k_{\alpha}^{(\mathsf{G})}(p^i, p^j) = e^{-\alpha \mathsf{G}(p^i,p^j)}\,; \tag{8.6}$$

$$w_4 = k_{\alpha}^{(\mathsf{E})}(p^i, p^j) = e^{-\alpha \mathsf{E}(p^i,p^j)}\,. \tag{8.7}$$

8.5.2 Multi-feature metrics to assess users similarity

As emerged in our preliminary study in Section 8.4, both user viewing direction and position on the virtual floor are relevant to detect similar behaviour among users. Therefore, the last set of proposed similarity metrics considers a combination of the above features. In detail, w_5 and w_6 are composed by the pairwise Euclidean distance in the virtual space $\mathsf{E}(x^i, x^j)$ and the difference of user relative distance to the volumetric content $||r^i - r^j||$ but include also the distance of their viewport centres projected on the volumetric content in terms of Geodesic distance $\mathsf{G}(p^i, p^j)$ and Euclidean distance $\mathsf{E}(p^i, p^j)$, respectively. More formally, we define the first:

$$\begin{aligned} w_5 &= k_{\alpha}^{(\mathsf{E})}(x^i, x^j) \cdot k_{\beta}^{(\mathsf{L})}(r^i, r^j) \cdot k_{\gamma}^{(\mathsf{G})}(p^i, p^j) \\ &= e^{-\alpha \mathsf{E}(x^i,x^j)} \cdot e^{-\beta ||r^i - r^j||} \cdot e^{-\gamma \mathsf{G}(p^i,p^j)}\,; \end{aligned} \tag{8.8}$$

while the second weight is equal to:

$$\begin{aligned} w_6 &= k_{\alpha}^{(\mathsf{E})}(x^i, x^j) \cdot k_{\beta}^{(\mathsf{L})}(r^i, r^j) \cdot k_{\gamma}^{(\mathsf{E})}(p^i, p^j) \\ &= e^{-\alpha \mathsf{E}(x^i, x^j)} \cdot e^{-\beta ||r^i - r^j||} \cdot e^{-\gamma \mathsf{E}(p^i, p^j)} . \end{aligned} \tag{8.9}$$

Our preliminary analysis has also highlighted a correlation between the viewport overlap of two users and their relative distance from the volumetric content. The closer users are to the volumetric content, the smaller and more detailed is the portion of the displayed content; the farther they are, the bigger but with fewer details becomes the displayed portion. Thus, in the first case, the high overlap between displayed areas of two different users is more difficult. To take into consideration this behaviour, we model the relative distance via a hyperbolic tangent kernel. Given the relative distance r_i between the user i and volumetric content, we evaluate it as follows:

$$\eta(r_i) = \tanh(r_i) . \tag{8.10}$$

As previously, metrics w_8 and w_9 are based on both user distance in the virtual floor $\mathsf{E}(x^i, x^j)$, and on the volumetric content in terms of Geodesic distance $\mathsf{G}(p^i, p^j)$ and Euclidean distance $\mathsf{E}(p^i, p^j)$, respectively. More formally, we define w_8 as following:

$$\begin{aligned} w_8 &= k_{\alpha}^{(\mathsf{E})}(x^i, x^j) \cdot \beta\big[\eta(r^i) + \eta(r^j)\big] \cdot k_{\gamma}^{(\mathsf{G})}(p^i, p^j) \\ &= e^{-\alpha \mathsf{E}(x^i, x^j)} \cdot \beta\big[\tanh(r_i) + \tanh(r_j)\big] \cdot e^{-\gamma \mathsf{G}(p^i, p^j)} ; \end{aligned} \tag{8.11}$$

while w_9 is:

$$\begin{aligned} w_9 &= k_{\alpha}^{(\mathsf{E})}(x^i, x^j) \cdot \beta\big[\eta(r^i) + \eta(r^j)\big] \cdot k_{\gamma}^{(\mathsf{E})}(p^i, p^j) \\ &= e^{-\alpha \mathsf{E}(x^i, x^j)} \cdot \beta\big[\tanh(r_i) + \tanh(r_j)\big] \cdot e^{-\gamma \mathsf{E}(p^i, p^j)} . \end{aligned} \tag{8.12}$$

8.6 Experimental setup

We now validate and test the above metrics using a point cloud dataset. First, we describe how we evaluate the performance of our similarity metrics. Then, we run an ablation study to evaluate for each similarity metrics the best performing set of regulators. The dataset that we test in the following is the same used for our preliminary analysis in Section 8.5. In the last column of Table 8.3, we provide the values of threshold per each similarity metric, evaluated as described previously in Section 8.5.

8.6.1 Performance Evaluation Setup

In order to test the validity of our proposed similarity metrics, we consider three performance metrics: averaged *overlap ratio* per cluster, *relevant clustered population*, and *precision*. The first two are more specific to our navigation trajectory in a 6-DoF system, while the latter is a popular index used to evaluate clustering algorithm performance.

Overlap ratio per cluster: as defined in Section 8.3.2, the overlap ratio computes the portion in common of displayed content between two users. Therefore, to compare the performance of our detected clusters with the different similarity metrics, we average the overlap ratio among all users who are put in the same group. More formally, given a detected cluster C_k is defined as follows:

$$O_k = \frac{1}{n_k} \sum_{\substack{i,j \in C_k \\ i \neq j}} O(i,j) \tag{8.13}$$

where i and j are two generic users, n_k is the cardinality of elements bellowing to clusters C_k, and $O(i,j)$ the overlap ratio as defined in Equation 8.1.

Relevant clustered population: the more users are clustered together with a high viewport overlap, the more meaningful are our clusters. Therefore, we consider as a relevant clustered population the sum of users that have been put in clusters with at least 2 other elements.

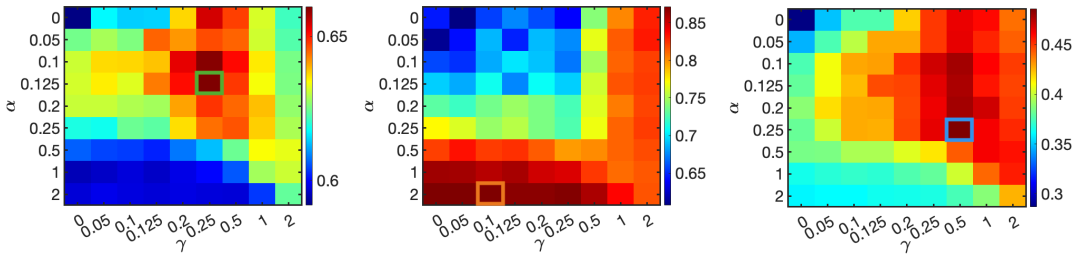
Precision: in a classification task, this index evaluates the portion of elements that are classified correctly and has values between 0 and 1 [221]. More formally:

$$P = \frac{TP}{TP + FP} \tag{8.14}$$

where True Positive (TP) (False Positive (FP)) is the number of viewers classified correctly (incorrectly) together in a cluster. In our case, two users are identified positively if they are in the same cluster and their viewport overlap is actually over the desired threshold.

8.6.2 Ablation Study

We now present an ablation study to tune the best set of regulator parameters that maximise the performance of each similarity metric. Equipped with the threshold values per each similarity metrics given in Table 8.3, we run a frame-based clustering to select the best set or sub-set of regulators α, β and σ. We test their performance in the following range of values $[0, 0.05, 0.1, 0.125, 0.2, 0.25, 0.5, 1, 2]$, based on navigation trajectories collected in the selected dataset above described; we used the performance metrics described in the previous section such as overlap



(a) Overlap Ratio **(b)** Relevant population **(c)** Precision

Figure 8.6: Example of parameter selection for w_7 with $\beta = 0.5$. Values set 1 selected based on max overlap, set 2 max clustered users, set 3 based on precision.

ratio, precision and relevant population. In detail, we average the final performance of clusters obtained by all similarity metrics over time and across content.

Single-feature metrics

For single-feature metrics ($w_1 - w_4$), we notice a very small variance in terms of performance per each similarity metric. Therefore, we decided to select $\alpha = 1$ for this set of metrics.

Multi-feature metrics

More challenging is instead the selection parameters for multi-feature metrics ($w_5 - w_8$). Each similarity metric depends on three parameters: α, β and γ. To overcome this, we first select three sets of parameters taking into account only navigation trajectories for reference content: one group of parameters (set 1) based on the maximum overlap ratio, the second (set 2) depending on the relevant clustered population and the last group (set 3) as the one reaching the highest precision. As an example, we report in Figure 8.6 the selection of these three sets of parameters for the similarity metric w_7. Then, we test these values on all the available trajectories included in the analysed dataset to finally select the best set of parameters. Table 8.4 provides all the performance of the multi-feature similarity metrics obtained by the three selected sets of parameters. Since there is no particular configuration that outperforms in terms of overlap ratio, relevant population and precision, we decided to select all regulators selected in set 3. This configuration indeed ensures a good balance of overlap ratio and relevant population for all the similarity metrics and ensure the highest value of precision. In Table 8.3, we summarise the selected regulator values used in the following analysis.

8.7 Analysis and Discussion

Equipped with the similarity metrics, the corresponding values of regulator parameters and threshold G_{th} reported in Table 8.3, we now conduct our validation study. In

Table 8.4: Parameter selections and their performance for multi-feature metrics (w_5 - w_8).

		w_5	w_6	w_7	w_8
set 1	$[\alpha, \beta, \gamma]$	[0.12, 0.125, 0.125]	[0.12, 1, 0.25]	[0.125, 0.5, 0.25]	[0.25, 0.5, 0.2]
	Overlap Ratio	0.63	0.64	0.66	**0.69**
	Relevant Population	0.82	0.78	0.69	0.62
	Precision	0.45	0.40	0.47	0.48
set 2	$[\alpha, \beta, \gamma]$	[1, 0.05, 0.05]	[0.5, 0.05, 0.05]	[2, 0.5, 0.1]	[2, 0.5, 0.05]
	Overlap Ratio	0.58	0.59	0.60	0.63
	Relevant Population	**0.91**	0.89	0.87	0.84
	Precision	0.32	0.32	0.36	0.33
set 3	$[\alpha, \beta, \gamma]$	[0.1, 0.5, 1]	[0.1, 0.125, 0.2]	[0.25, 0.5, 0.5]	[0.5, 0.5, 0.5]
	Overlap Ratio	0.63	0.63	0.65	0.66
	Relevant Population	0.83	0.80	0.77	0.74
	Precision	0.45	0.44	**0.49**	0.48

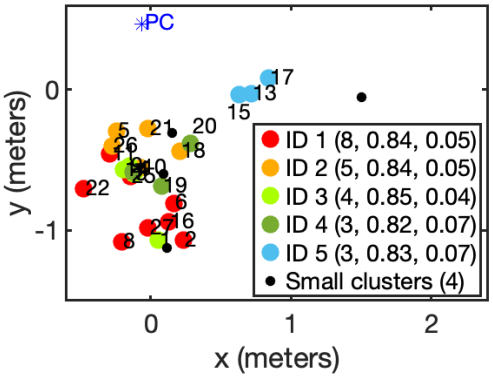
detail, we focus on analysing navigation trajectories experienced with non-distorted content.

8.7.1 Frame-Based Analysis

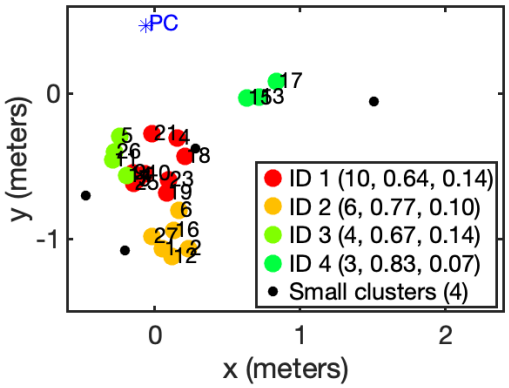
As first step, we implement a frame-based analysis (*i.e.*, frame-based clustering) to visually verify the performance of the detected clusters in the different settings of similarities. Figure 8.7 shows the clusters detected using the ground-truth metric O to construct the graph (Figure 8.7 (a)) with the ones given based on each proposed single-feature similarity metric (Figure 8.7 (b-e)), for frame 50 of sequence PC1. Similarly, Figure 8.8 shows results based on the proposed multi-feature metrics. In particular, in both set of figures, each user is represented by a point on the VR floor which is coloured based on the assigned ID cluster, whereas the volumetric content is symbolised by a blue star. For each relevant cluster (*i.e.*, cluster with more than 2 users), we provide in the legend the following results: number of users inside the cluster, the average and variance of the overlap ratio among all users within the cluster. Finally, we represent the remaining users which are in either single or couple-cluster as black points; the total number of these users is also provided in the legend as "Small clusters (total number of non-relevant clusters)". We can notice that single-feature metrics, Figure 8.7 (b-e), have the tendency to create very populated clusters but with a low overlap ratio. For instance, w_3 and w_4 generate a main big cluster with 18 and 19 users, respectively, while the corresponding overlap ratio drops drastically to 0.62. The only exception is given by w_1, which generates a variable set of clusters with consistent values of overlap ratio, over 0.64. Let us now consider as an example the users 13, 15 and 17, which in the ground-truth case (Figure 8.7 (a)) form their own cluster (*i.e.*, ID 5) with a high overlap ratio (0.83), and user 24, who is quite isolated from other users and belongs to a single cluster. We can notice that w_2 and w_4 fail in detecting the group of users 13, 15 and 17 as similar, dividing them instead in different clusters. On the other hand, w_3 detects this similarity but puts user 24 in a relevant cluster (ID 1). From these observations,

we can notice that the projection of the viewport centre on the volumetric content, which forms the basis of w_3 and w_4, is not sufficient to correctly identify similar users. Analogously, considering only the difference in terms of the relative distance between the user and volumetric content, as done in w_2, does not allow to detect similarity among users. Thus, the most promising metric in this group seems to be w_1, which is based on the user position on the virtual floor. The last group of Figure 8.8 shows clusters based on multi-feature similarity metrics. In all these settings, a total of four main clusters are detected, except for w_6 that leads to three clusters, as shown in Figure 8.8 (b). The latter detects the highest number of small clusters (6) while being the only one that does not identify users 13, 15 and 17 as belonging to the same cluster. On the contrary, the other three metrics w_5, w_7 and w_8 detect a main cluster and three smaller clusters with a consistent overlap ratio. For instance, the resulting clusters based on w_5 have an overlap ratio always bigger than 0.69 and only two users fall in small clusters. Overall, multi-feature metrics appear to better suit for detecting similar users than previous ones, except for w_6. This is expected as higher degrees of freedom are given to users, the more challenging is the system, and thus users similarity to detect.

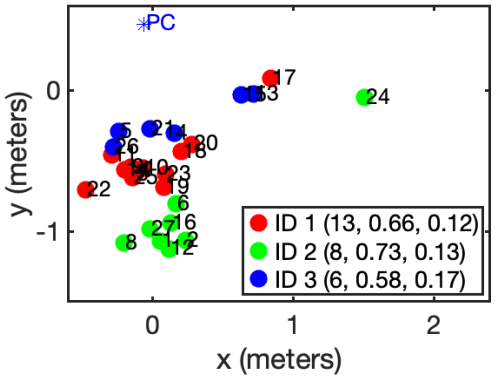
Instead of looking at one frame only, we now analyse the per-frame clustering technique providing in Table 8.5 the performance averaged over time and corresponding standard deviation. In detail, we show the average and standard deviation of performance metrics described in Section 8.6 across the entire analysed dataset. To be noted that clusters based on w_2 are able to group in the relevant clusters the majority of the population in all the analysed PCs (reaching the maximum value of 0.94 in PC1) to the detriment of precision, which falls to values between 0.22 and 0.35. As already shown in the previous investigation, the most promising similarity metrics in terms of precision and overlap ratio are both w_7 and w_8 followed by w_5. These outperform the other weights in all PCs, ensuring an overlap ratio within the same cluster with values in the range of 0.59 and 0.70 for w_7, 0.60 and 0.72 for w_8. Similarly, the values of precision are always over 0.42 for both w_7 and w_8. The only exception is in PC1, where the best performing metric in terms of precision is w_6, which for the other contents cases is always the worst performing among multi-functional metrics.



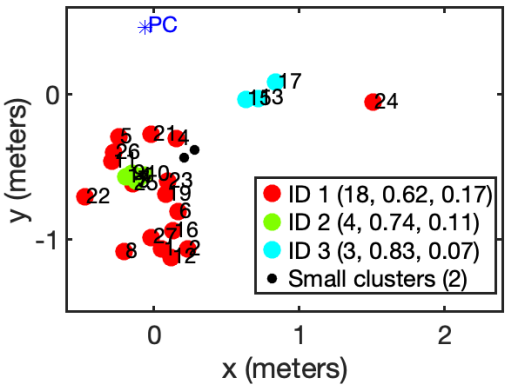
(a) Ground-truth (O_{th} = 75%)



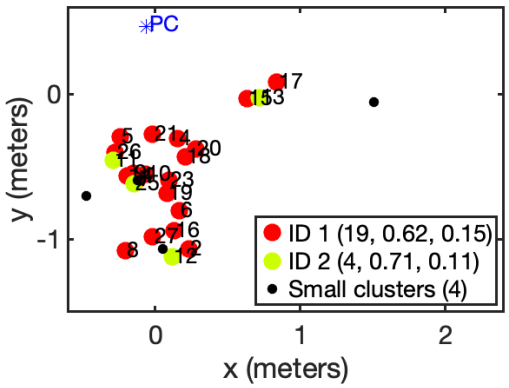
(b) w_1 (single feature metric)



(c) w_2 (single feature metric)

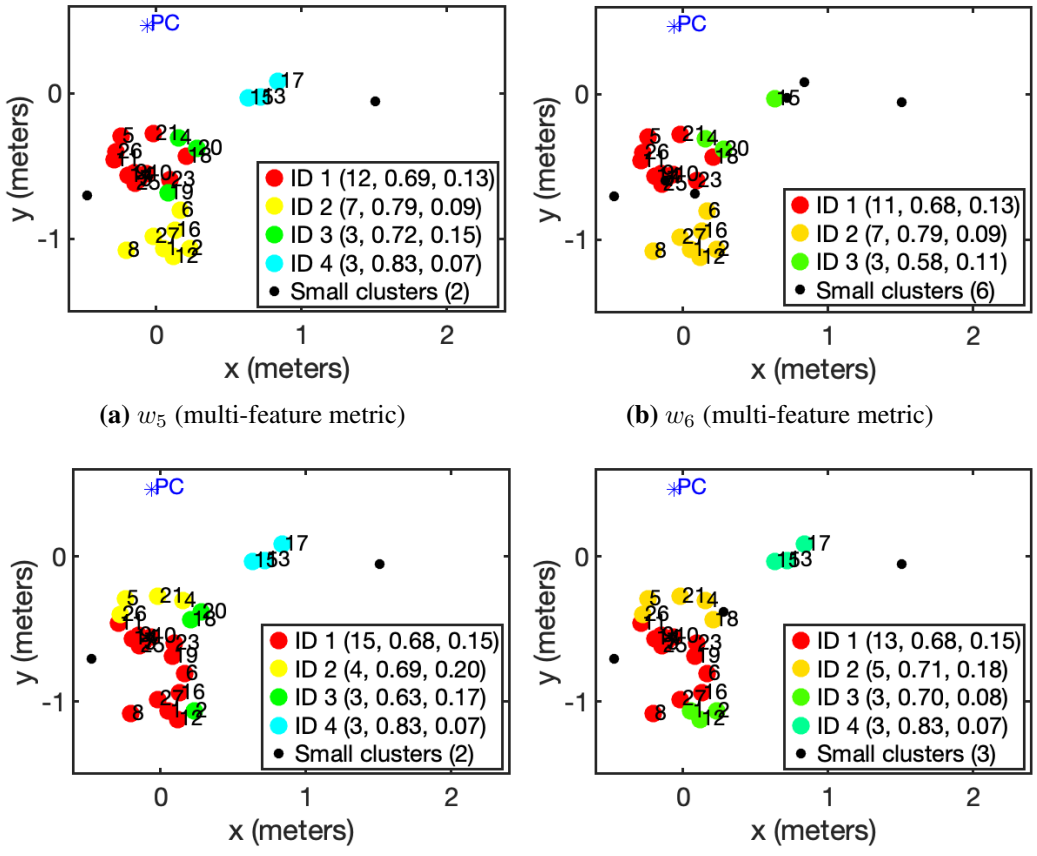


(d) w_3 (single feature metric)



(e) w_4 (single feature metric)

Figure 8.7: Cluster results in frame 50 of sequence PC1 (*Longdress*) per single-feature metrics. Each dot represents a user on the virtual floor while the blue star stands for the volumetric content. In the legend, per each cluster with more than 2 users are reported on brackets the following values: the number of users included in the same cluster, averaged pairwise viewport overlap and corresponding variance within the cluster.



(a) w_5 (multi-feature metric) **(b)** w_6 (multi-feature metric)

(c) w_7 (multi-feature metric) **(d)** w_8 (multi-feature metric)

Figure 8.8: Cluster results in frame 50 of sequence PC1 (*Longdress*) per multi-feature metrics. Each dot represents a user on the virtual floor while the blue star stands for the volumetric content. In the legend, per each cluster with more than 2 users are reported on brackets the following values: the number of users included in the same cluster, averaged pairwise viewport overlap and corresponding variance within the cluster.

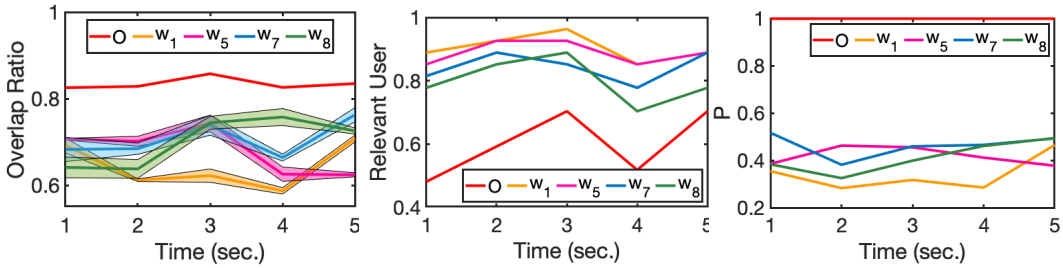
Table 8.5: Results in terms of averaged and standard deviation per each performance metric across the entire dataset.

	Metrics	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
PC1	Overlap Ratio	0.68 ± 0.05	0.65 ± 0.04	0.66 ± 0.04	0.68 ± 0.07	0.70 ± 0.05	0.71 ± 0.05	0.70 ± 0.05	$\mathbf{0.72 \pm 0.06}$
	Relevant Population	0.85 ± 0.04	$\mathbf{0.94 \pm 0.03}$	0.92 ± 0.05	0.84 ± 0.08	0.83 ± 0.06	0.83 ± 0.07	0.83 ± 0.06	0.83 ± 0.07
	Precision	0.44 ± 0.06	0.35 ± 0.05	0.39 ± 0.07	0.30 ± 0.06	0.47 ± 0.07	$\mathbf{0.49 \pm 0.08}$	0.46 ± 0.07	0.44 ± 0.10
PC2	Overlap Ratio	0.57 ± 0.08	0.53 ± 0.09	0.54 ± 0.12	0.54 ± 0.11	0.59 ± 0.08	0.58 ± 0.08	0.59 ± 0.12	$\mathbf{0.60 \pm 0.10}$
	Relevant Population	0.80 ± 0.07	$\mathbf{0.92 \pm 0.06}$	0.83 ± 0.07	0.89 ± 0.06	0.80 ± 0.10	0.81 ± 0.07	0.72 ± 0.08	0.73 ± 0.06
	Precision	0.45 ± 0.06	0.28 ± 0.08	0.31 ± 0.08	0.27 ± 0.08	0.47 ± 0.09	0.42 ± 0.08	$\mathbf{0.54 \pm 0.08}$	$\mathbf{0.54 \pm 0.12}$
PC3	Overlap Ratio	0.65 ± 0.06	0.60 ± 0.07	0.64 ± 0.05	0.68 ± 0.06	0.65 ± 0.06	0.65 ± 0.06	0.68 ± 0.05	$\mathbf{0.69 \pm 0.05}$
	Relevant Population	0.82 ± 0.07	$\mathbf{0.93 \pm 0.05}$	0.88 ± 0.06	0.82 ± 0.08	0.84 ± 0.06	0.81 ± 0.07	0.72 ± 0.07	0.70 ± 0.07
	Precision	0.48 ± 0.11	0.36 ± 0.08	0.39 ± 0.07	0.39 ± 0.06	0.49 ± 0.11	0.49 ± 0.10	0.52 ± 0.08	$\mathbf{0.55 \pm 0.08}$
PC4	Overlap Ratio	0.60 ± 0.04	0.52 ± 0.06	0.55 ± 0.03	0.59 ± 0.06	0.59 ± 0.04	0.58 ± 0.05	0.61 ± 0.04	$\mathbf{0.66 \pm 0.05}$
	Relevant Population	0.82 ± 0.07	$\mathbf{0.92 \pm 0.05}$	0.90 ± 0.08	0.86 ± 0.08	0.83 ± 0.08	0.77 ± 0.07	0.80 ± 0.07	0.71 ± 0.08
	Precision	0.35 ± 0.06	0.22 ± 0.04	0.31 ± 0.06	0.25 ± 0.07	0.38 ± 0.07	0.38 ± 0.09	$\mathbf{0.42 \pm 0.06}$	$\mathbf{0.42 \pm 0.07}$
All PCs	Overlap Ratio	0.62 ± 0.06	0.57 ± 0.06	0.60 ± 0.06	0.62 ± 0.07	0.63 ± 0.06	0.63 ± 0.06	0.65 ± 0.06	$\mathbf{0.66 \pm 0.06}$
	Relevant Population	0.82 ± 0.06	$\mathbf{0.93 \pm 0.05}$	0.88 ± 0.07	0.85 ± 0.08	0.83 ± 0.07	0.80 ± 0.07	0.77 ± 0.07	0.74 ± 0.07
	Precision	0.43 ± 0.07	0.30 ± 0.06	0.35 ± 0.07	0.30 ± 0.07	0.45 ± 0.09	0.45 ± 0.09	$\mathbf{0.49 \pm 0.07}$	0.48 ± 0.09

8.7.2 Trajectory-Based analysis

Given the above remarks, we now analyse the performance metrics over time, taking into account only w_1, w_5, w_7, and w_8. Indeed, we decide to select the best performing similarity metrics in the previous investigation (w_5, w_7 and w_8). To have a fair comparison, we also keep the most promising among the single-feature metrics, w_1. We compute clique-based clusters over a time window of $1s$ (*i.e.*, chunk) and a time similarity threshold of $0.8s$. At each chunk, we evaluate the average overlap ratio per relevant cluster (*i.e.*, cluster with at least two elements), the average of the relevant population and the precision of detected clusters. As an example, we show these results per sequence PC1 (*Longdress*) as functions of time for each similarity metric under consideration in each subplot of Figure 8.9. In the figures, we also add the performance of clusters detected by the ground-truth metric O (*i.e.*, red line). We observe that all similarity metrics reach an average overlap ratio within clusters between 0.6 and 0.75 (Figure 8.9 (a)). However, clusters based on w_1 appear to have lower performance than other metrics which are quite similar, although with a slight predominance of w_7. In terms of relevant users (Figure 8.9 (b)), it is worth noting that all the proposed similarity metrics generate bigger clusters than the ground-truth metric, which considers only half of the population as relevant. In more detail, the clusters resulting from w_1, w_5 and w_8 put in relevant clusters 0.8 of the entire population for all the sequence time. Finally, in terms of precision as highlighted in Figure 8.9 (c) the only similarity metric that generated clusters with P over to 0.4 in the entire sequence is w_7. These investigations show that similarity metrics based on multi-feature, such as w_7 and w_8, are more promising for detecting users with a similar behaviour while experiencing volumetric content.

In summary, from this validation analysis, we can conclude the following:



(a) Mean Overlap Ratio in Relevant Cluster. **(b)** Mean Relevant Users. **(c)** Precision.

Figure 8.9: Spherical clustering over time (chunk = 1 sec.) results per sequence PC1 (*Long-dress*): comparison between Ground-truth, and a subset of proposed metrics (w_1, w_5, w_7 and w_8).

- overall, *multi-feature metrics* are more precise in detecting users with similar behaviour (in terms of displayed content) both in a frame- and chunk-based analysis;
- in particular, in spite of the slightly more complex formulation, w_7 and w_8 are robust and easy-to-use metrics that ensure a robust and reliable behavioural analysis via clustering tools;
- on the contrary, metrics based only on a single feature (*i.e.*, *single-feature metrics*) are not sufficient to correctly identify similar users;
- the only exception among single-feature metrics is w_1 which is based only on the position of the user on the floor. Despite its simplicity, this metric is comparable with multi-feature metrics. Hence, it can be used for an easy-to-implement preliminary behavioural analysis.

These considerations have been built on point clouds with human body. We leave further investigations across multiple types of content for future works.

8.8 Case study

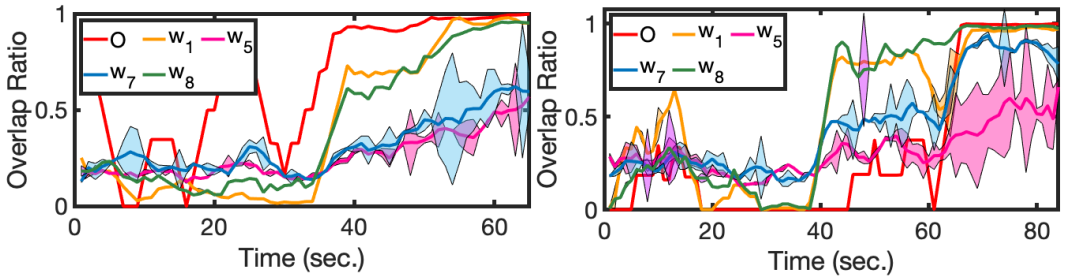
The above study has been carried out with a dataset only and we are therefore interested in understanding if insights from the above study could be applied to other human-like datasets. To show that our study generalises to datasets, we now investigate the proposed metrics on the dataset presented in [208]. Authors have collected navigation trajectories of 20 users while displaying volumetric content in an Augmented Reality (AR) scenario. Similarly to the previously analysed dataset presented in Section 8.4, a single object of interest was placed in the scene. Specifically, two dynamic volumetric human body sequences represented as 3D meshes

(a) *Nico* (VV1) **(b)** *Sir Frederic* (VV2)

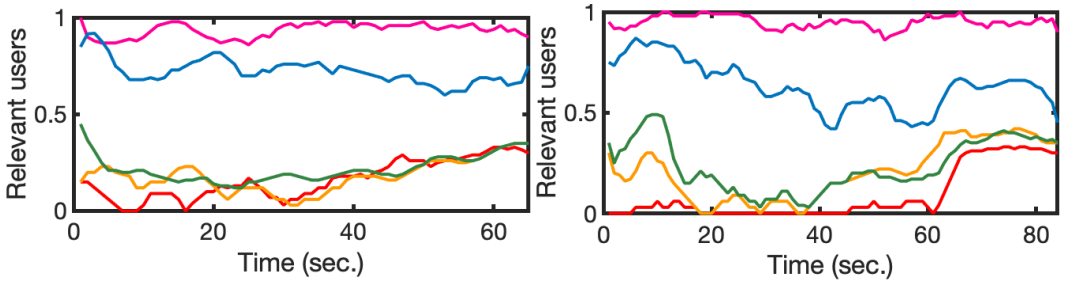
Figure 8.10: Volumetric Human Body sequences used in the AR dataset analysed in our case study.

with texture information were used: *Nico* (VV1) and *Sir Frederic* (VV2) in Figure 8.10, respectively. In order to conduct our study, both the sequences were kindly made available by Volograms upon request [66, 222]. The navigation data have been collected in a remote scenario through an Android AR application, which allowed users to display the volumetric content from any desired location and portable device (*e.g.*, smartphone). Participants were also free to display the volumetric content how they most preferred. Thus, the main differences with the previously analysed dataset are the following: the different format of volumetric content (3D mesh instead of point cloud), different immersive scenario (AR instead of VR application) and the heterogeneity of viewing devices (any smartphone device instead of a specific HMD). In particular, the 3D mesh content does not allow for a simple formulation of the overlap ratio as we have described it in Section 8.3.2. For consistency, we convert the sequences from 3D meshes to point clouds by discarding edge information and only keeping vertices as points; we discuss the inherent challenges to define our ground-truth metric in Section 8.9.

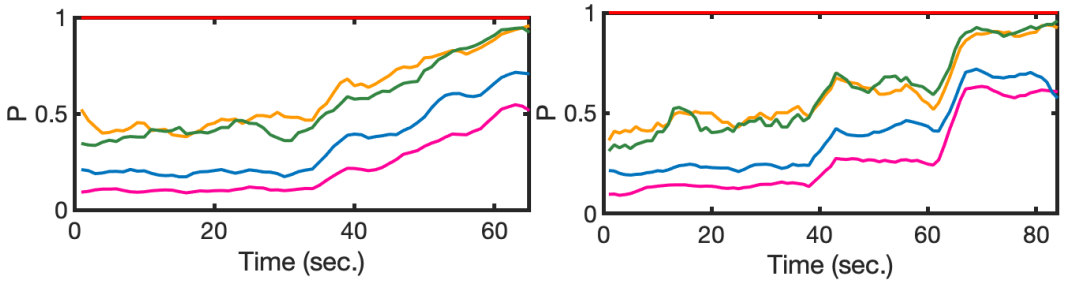
Similarly to our previous investigations, we now apply to this new scenario the spherical clustering based on the subset of best-performing feature metrics, such as w_1, w_5, w_7 and w_8. We evaluate clusters in chunks of length $1s$ with a time similarity threshold of $0.8s$ and the threshold values G_{th} reported in Table 8.3. At each chunk, we compute the average overlap ratio per relevant cluster (*i.e.*, cluster with at least two elements), the average of the relevant population and the precision



(a) VV1 - Mean Overlap Ratio in Relevant Cluster. **(b) VV2** - Mean Overlap Ratio in Relevant Cluster.



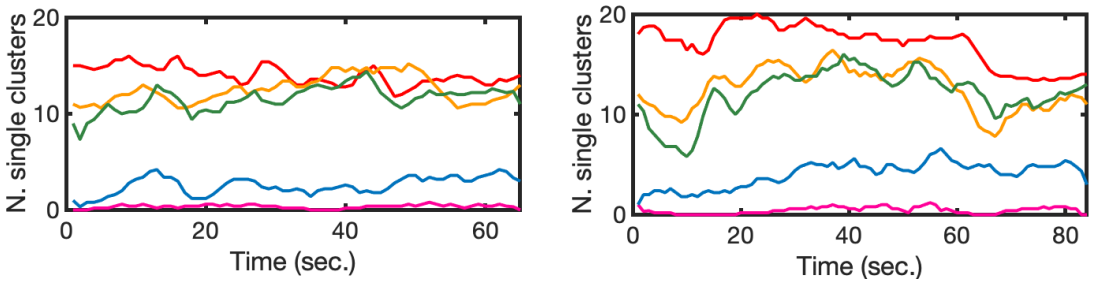
(c) VV1 - Mean Relevant Users. **(d) VV2** - Mean Relevant Users.



(e) VV1 - Precision. **(f) VV2** - Precision.

Figure 8.11: Spherical clustering over time (chunk = 1 sec.) results per sequence VV1 (*Nico*) and VV2 (*Sir Fredrick*): performance comparison between ground-truth, and a subset of proposed metrics (w_1, w_5, w_7 and w_8).

of the detected clusters. Figure 8.11 shows these results as a function of time per each selected similarity metric, in particular, the first row refers to *Nico* (VV1) while the second one to *Sir Frederic* (VV2). Since viewers were allowed to drop the AR experience at any desired time, in the following we consider only the time window in which 75% of the user population (15 out of 20 viewers) are still in the experiment: 63 and 83 seconds, respectively for VV1 and VV2. We observe that both the sequences have an initial moment of adjustment where viewers are displaying different portions of the content. This is detected by clusters based on the overlap ratio (*i.e.*, red line) which do not have a consistent pairwise overlap. For instance, Figure 8.11 (a) shows in the first $40s$ of the immersive experience for VV1 the average of overlap ratio within the main detected clusters has up and down for all the metrics, included the ground-truth. However, this behaviour stabilises around



(a) VV1 **(b)** VV2

Figure 8.12: Single-user cluster per sequence VV1 (*Nico*) and VV2 (*Sir Fredrick*) obtained via spherical clustering based on overlap ratio, and a subset of proposed similarity metrics (w_1, w_5, w_7 and w_8).

$40s$ when the overlap ratio for the ground-truth metric converge to 1. Similarly, the performance metric detected by w_1 and w_8 reaches values above 0.6 with a very low variance for both the metrics. On the contrary, the overlap ratio of cluster detected by w_5 and w_7 has values below 0.5 and a more consistent variance over time. Indeed, these metrics in terms of relevant users (Figure 8.11 (c)) generate bigger clusters compared to the ground-truth metric. In particular, w_5 considers almost the entire population at any given time in a big cluster, which is quite the opposite behaviour to the ground-truth metric. This metric indeed generates small relevant clusters most of the time; clusters based on w_1 and w_8 follow a very similar trend. These two metrics are also the best performing in terms of precision, as shown in Figure 8.11 (e) with values always above 0.4. A similar general behaviour is also observable for VV2 in the second column of Figure 8.11. In this volumetric content, users explore more randomly during the first minute of the experience to end up having similar behaviour, in fact, clusters have a consistent overlap ratio (Figure 8.11 (b)). As in the previous example, similarity metrics w_1 and w_8 are more precise in reflecting the ground-truth behaviour and thus, detecting viewers with similar behaviour and putting them within the same clusters. Finally, it is worth noticing that between 40 and 60 seconds, all the similarity metrics reach a higher overlap ratio compared to the ground-truth performance, in particular, metrics w_1 and w_8. This opens new questions to be further investigated as discussed in the following section.

8.9 Discussion

In this chapter, we have presented the main challenges of user behavioural analysis in a 6-DoF system due to the new settings and the added locomotion functionalities. Since behavioural analysis of 6-DoF users is not considered in the literature yet, there is no reference metric available to detect viewers who are displaying the same

portion of the content. Thus, we had to define a general ground-truth user similarity metric, such as *overlap ratio*. To be as general as possible, we established the overlap as the percent of points displayed in common by two users. This is fairly straightforward, albeit time-consuming, to compute for point cloud contents, in which each point is rendered separately. For other types of volumetric contents, determining the overlap ratio is not as simple. Considering the number of vertexes that fall into a given frustum could lead to misleading results when large faces between sparsely distributed vertexes are present. Moreover, the metric requires to render each volumetric video at any given time and for each viewer, making its computation not trivial and intensely time-consuming. To overcome this issue and to assess users similarity in a simple and objective way, we investigate several similarity metrics considering different distance features and measurements. In detail, we investigate different features or combinations of them which consider users location in the virtual space and their viewing direction. First, we validate and test our similarity metrics via a clique-based clustering tool proposed for 3-DoF scenario on real navigation trajectory collected in a 6-DoF VR environment. Our extensive analysis shows that metrics based on multi-features better model and thus, detect similarity among users, reaching encouraging values of both overlap ratio and precision. Therefore, we tested their performance on a different kind of 6-DoF navigation trajectories. In this second dataset, viewers displayed volumetric content in an AR scenario through smartphones. Therefore, even if users were enabled with the same 6-DoF locomotion settings, the viewing device and the FoV were different. Despite these differences, our proposed similarity metrics are still good at identifying viewers who are displaying similar content. In this context not only a multi-feature metric is outperforming the others, but also the simplest one which is based only on the users location in the virtual space. This opens the gate to further investigations aimed at detecting user behavioural differences in a 6-DoF experience done in VR and AR settings. These are indeed essential to be exploited in efficient user-centric solutions for both immersive systems. Finally, it is worth mentioning that our ground-truth metric of similarity is very tight in detecting similar users, especially in an AR scenario. As an example, Figure 8.12 shows the number of single clusters detected over time by the overlap ratio (*i.e.*, red line) and the sub-set of most performing similarity metrics for both the volumetric sequences of the second analysed dataset. In particular, in VV2 (Figure 8.12 (b)), the clique-based clustering based on the overlap ratio does not detect similar users such that the majority of the population are put in a single cluster. Therefore, further analysis is needed to test if in this scenario a different overlap threshold better model similarity among users.

8.10 Conclusion

In this chapter, we have presented similarity metrics to enable behavioural analysis of users while exploring a 6-DoF immersive content. We were interested in modelling similarities among users *observing the same volumetric content*. In our first attempt of behavioural analysis in 6-DoF, we have shown that the way in which users interact within a 3- and 6-DoF scenario is fundamentally different preventing a straightforward extension of current 3-DoF algorithms to 6-DoF. Therefore, we advanced the state-of-the-art, proposing novel similarity metrics taking into account the new physical settings and locomotion functionalities given to users. Our results showed that solutions that consider both user position and viewing direction are promising to correctly detect users with a similar behaviour while experiencing volumetric content. We have also demonstrated the robustness and versatility of these metrics, which preserve good performance on navigation trajectories collected in a 6-DoF AR scenario.

Chapter 9

Behavioural analysis in a social VR crime movie

In this chapter, a case study on behavioural analysis of user navigating in 6-DoF social VR movie is presented. Social Virtual Reality (VR) applications represent a big step forward in the field of remote communication other than providing the possibility for participants to explore and interact with virtual environments and objects, allowing them of being together in the same virtual space. In the following, we conduct an investigation on how users are affected by virtual characters and narrative elements of the movie through objective metrics, showing a more static behaviour when an interactive task was requested, and more exploratory movements during dialogues.

9.1 Introduction

Virtual Reality (VR) applications are going through a rapid evolution of technology, getting integrated in daily-life devices such as smartphones and laptops. Therefore, it is possible to imagine that in the near future video calls will give users a completely different experience than now: people will be able to chat, walk together, interact with virtual objects and watch events such as concerts or movies together in a common virtual environment [223]. This is what *social VR* applications are pledging to enable, becoming a promising tool of the near future remote communications [224]. VR applications overpassed the traditional paradigm of passively consuming multimedia content, enabling a sense of immersiveness and interaction by placing the users at the centre of the action; a further step toward fully immersive services has been attempted by social VR. This emerging remote communication tool is indeed aiming at overstepping current remote communications through 2D screens, enabling instead virtual co-presence of more users within the same virtual environment and allowing body interactions similarly to face-to-face communication

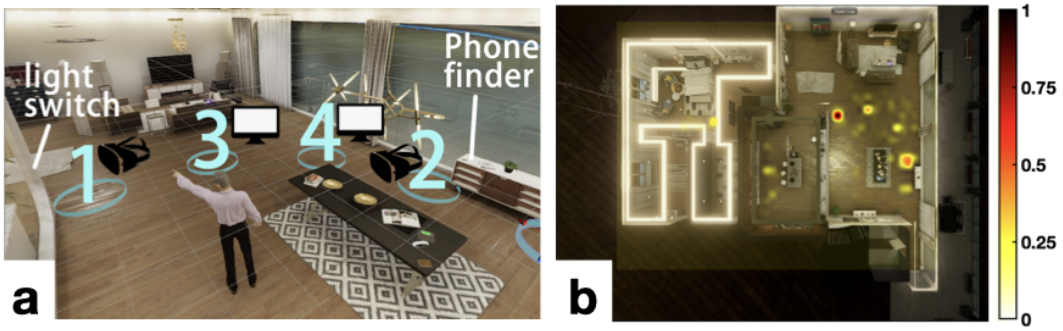


Figure 9.1: a) Living room with indicated the starting position of each user: 1 and 2 are HMD users while 3 and 4 are desktop users. There are also two interactive objects (*i.e.*, the light switch on the left and phone finder on the right) and the main virtual character, detective Sarge. b) Floor map of the virtual house with the user heatmap of main locations visited over time.

[224–226]. The new challenge is therefore to increase the realism in the virtual experience and interaction. Therefore, the key aspect that needs to be fully understood in order to advance in this technology is the user, the main director of the virtual experience.

Emerging Social VR platform, such as *Facebook Horizon*[1] and *Mozilla Hubs*[2], are rapidly growing in popularity. For instance, the latter one has been used in many recent academic events (*i.e.*, conferences: IEEE VR 2020[3], ACM IMX 2020[4], QoMEX 2020[5] and ACM CHI 2020 Social VR workshop [224]). In most of these applications, participants interact among each other by taking part of 3D virtual spaces through a computer-generated and customised avatar. Physical displacement and proxemic interactions in virtual environments have been analysed to investigate which social cues are the most influencing and therefore are needed to ensure presence and immersion [227, 228]. Moreover, many works have investigated the advantages to have a more realistic self-representation versus a 3D avatar providing a higher degree of immersion and presence in the VR experience [229, 230]. Therefore recently, a more natural self-representation of participants has been introduced, thanks to real-time acquisitions and reconstructions of point clouds by depth cameras [70]. Not to be neglected is the technological aspect of these new applications. Social interactions and photorealistic representations come with both computational and bandwidth overhead for transmission and rendering [71]. Analysing user behaviour in terms of viewing angle and head trajectories, among other objective measurements, represents the first step towards the building of real-time and

[1] https://www.oculus.com/facebook-horizon/
[2] https://hubs.mozilla.com
[3] https://ieeevr.org/2020/
[4] https://imx.acm.org/2020/
[5] https://www.qomex2020.ie

realistic VR systems at large that can optimise delivery based on user-centred adaptation [92, 96].

In this chapter, we focus on a better understanding of how people interact with a virtual environment and other users within it. In details, we present a first attempt of objective behavioural analysis in a 6-DoF social and interactive VR movie. We based our analysis on navigation trajectories collected in a novel type of interactivity crime movie: four users, either equipped by HMD or desktop computer and a controller, were watching together a VR crime solving movie [231]. Figure 9.1 (a) show a snapshot of the living room in which the story mainly takes place, with its main virtual character. Using hyper-realistic self-user representations and enabling activities such as handling virtual objects and talking with characters, the main novelty of this experience is to provide a unique sense of immersion and social connectedness going beyond to a collaborative game experience [231]. We selected this content because highly representative of social VR content, in which users are free to move in the 3D space (6-DoF), but also to interact among themselves within a guided content. These features however lies key challenges that we aim at overcoming in this work. As first we need to consider both the new physical settings and locomotion functionalities given to users in a 6-DoF system. The user now not only can select the portion to be displayed by rotating the head as in a 3-DoF system but can also move inside the virtual environment changing the distance and perspective with the displayed content. The second challenge to consider is the guided and interactive behaviour that is the new social and interactivity features of the application that brought an added level of dynamics. For instance, during the experience participants are asked to make simple tasks, such as to look for and to press a button; their action influences directly the narration of the story since the time to solve the task is not fixed. Therefore, we compare user behaviour in terms of spatial displacements in the virtual environment and viewing direction with respect to movie characters and other participant within the whole experience. In particular, we show how much narrative elements of the movie, such as virtual characters movements or request of interactions, influence user behaviour.
In the following, we give more details about the crime movie and experimental setup in Section 9.2. Section 9.3 presents the core of our analysis and discussions on our results highlighting key aspects that influence the user's behaviour in the social VR experience. Finally, a summary of our work and findings is provided in Section 9.4.

9.2 A social VR Murder Mystery Movie

We contextualise our experiment in a social VR setting, in which 4 users are called to experience an immersive movie, occasionally being asked to take part of the

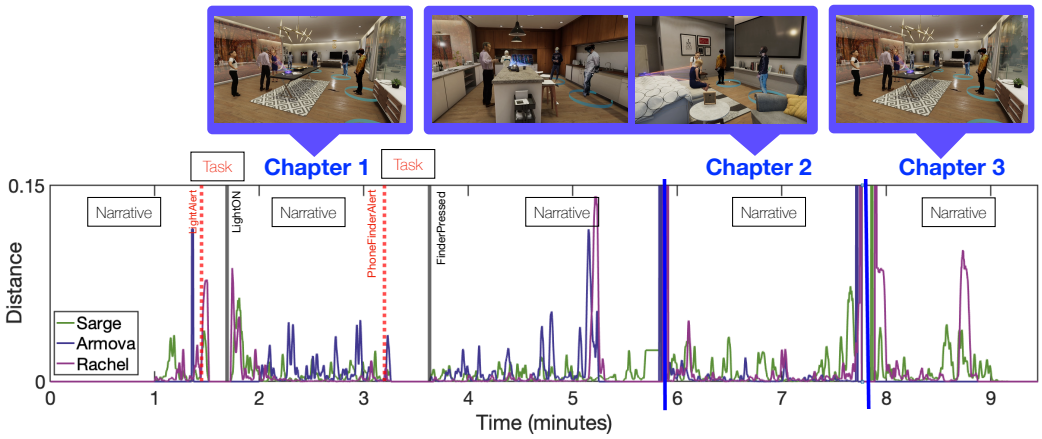


Figure 9.2: Timeline of the VR murder mystery movie, and spatial movements of virtual characters over time. Distance is computed with respect to the position in the previous frame. A screenshot per each chapter is also reported.

story. The scenario allows for users with 6-DoF navigation within a photorealistic 3D environment, which is populated by three virtual characters. In the following, we describe the movie timeline, the setup used during experiments and finally, we show some general performance of the system.

9.2.1 Movie plot

The interactive and immersive VR movie used in the experiments is fragment of a murder mystery investigation [231], led by detective Sarge Hoffsteler and his assistant, Rachel Tyrell. The users (4 in total) help the investigation. The victim is Elena Armova, who lived in a luxury apartment in central London. The story is split into 3 chapters, as depicted in Figure 9.2. In Chapter 1, the 4 users are placed in the virtual living room of the victim Ms. Armova, adopting an initial fixed positions indicated in Figure 9.1 (a). Participants mainly listen to a rendered victim interrogation, which is possible thank to a futuristic machine based on artificial intelligence. There are also two moments in which users are asked to interactively interact with objects in the scene: *(1)* user 1 is asked to switch on the light, *(2)* user 2 has to pick up a phone finder controller and press the button. This split Chapter 1 into three narrative moments where mainly virtual characters are talking and walking around the scene (*narrative* label in Figure 9.2) interleaved with two moments of active tasks for the user (*task* label). At the end of Chapter 1, users are split into two groups: user 1 and 3 are conducted to the virtual kitchen with detective Sarge, whereas user 2 and 4 are led to the virtual bedroom with Rachel and Ms. Armova. In both rooms, participants are listening to narrative dialogues of virtual characters. At the end of Chapter 2, the users are brought back together in the virtual living room for the final chapter, where detective Sarge describes how

the murder has been solved.

9.2.2 Experimental Setup

A low-latency volumetric video delivery pipeline, based on point cloud representation, was used to place each participant into the virtual scene [70]. Each user was captured using 3 Kinect Azure devices, placed in a circle around them, 120° apart from each other. This allowed each participant to be captured from multiple angles, ensuring a photorealistic representation while they interacted with the scene.

Two devices were used to visualise the social VR experience: users 1 and 2 were equipped with Oculus Rift HMDs, complete with controllers, whereas users 3 and 4 could watch the scene through 50-inch monitors, and could navigate using gaming joysticks. For HMD users, teleportation was enabled in key locations of the scene, as physical locomotion was restricted due to the acquisition setup, whereas for desktop users, movement was enable through the gaming joystick. Due to the configuration of the controllers, only HMD users were able to engage with the interactive elements in the scene.

A total of 48 participants was recruited for the experiment, resulting in 12 social VR sessions. The number of users was selected to ensure at least 24 participants per condition (HMD vs desktop). The sample size was determined using software G*Power [232], considering the between-subject design, assuming a large effect size ($d = 0.7$) and setting $\alpha = 0.05$ and desired power $1 - \beta = 0.75$. The participants were between 21 and 56 years old ($\mu = 34.9, \sigma = 10.3$). The gender distribution was balanced (23 males, 25 females). Users were randomly assigned to each device hence to the initial position of the experiment. All of them were fluent in English, and had no motor or visual impairment. Participants knew themselves always in advance, in detail there were always at least two groups of friends or relatives per each experiment.

Before and after the virtual experience, semi-structured interviews were also conducted to collect explicit feedback from users. Analysing these explicit feedback is however beyond the scope of this paper, which is rather focused on building new metric to analyse users behaviour and deduce implicit feedback. Each experimental session lasted approximately 60 minutes and consisted of the following main parts: **Part 1** *(10 minutes)*: explanation of the experiment including main goal, procedure and description of the movie characters. This phase includes filling-in questionnaire too; **Part 2** *(10 minutes)*: training phase to let participants familiarise with devices and to interact with each other and virtual object within the virtual environment; **Part 3** *(10 minutes)*: virtual movie experiment; **Part 4** *(20 minutes)*: questionnaire phase to evaluate the social VR experience (*e.g.*, filling of presence/immersion, vi-

Figure 9.3: User motion based on both spatial and rotation movements per each session in the movie.

sual quality). **Part 5** *(10 minutes)*: a semi-structured group interview with all 4 participants. During the sessions, the position and rotation of the camera objects associated with each users were recorded at 30 Hz. From the logged data it was possible to compute the latency between encoding and rendering, for each device under use. The data was obtained at the granularity of one second, to avoid disrupting the performance of the system. On average, the point count for each frame was 86342 points per cloud (25 percentile: 82412.5; median: 90268; 75 percentile: 95727.5). The observed framerate for all the representations was on average 9.08 frames per second (25 percentile: 7.5; median: 8.9; 75 percentile: 10.6). The observed latency in all sessions was generally lower than 1 second, and was remarkably smaller for self-representation. In particular, mean latency for self-representation was 0.1147 seconds (25 percentile: 0.079; median: 0.105; 75 percentile: 0.136), whereas for the rest of the cases, it amounted to 0.5526 seconds (25 percentile: 0.408; median: 0.538; 75 percentile: 0.689).

9.3 Behavioural analysis

In the following, we first present a general analysis of users movements within the entire virtual environment and movie. In order to better understand how user behaviour changed over time as the movie progressed, we also analyse their position and orientation with respect first to the virtual characters (avatars) and finally, to other participants.

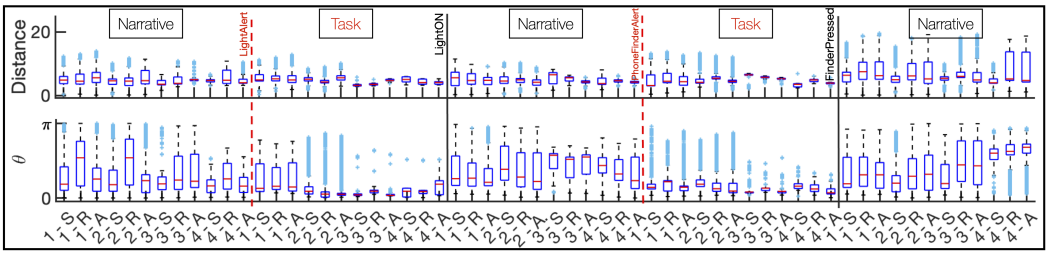
General movements analysis. A general overview of the exploration behaviour of users is given in Figure 9.1 (b), which shows a heatmap of the most visited locations in the virtual house, obtained by aggregating all the position data collected in the experiment. As described in Section 9.2.1, large part of the movie takes place in the living room (Chapter 1 and 3), which is reflected in the figure. The most visited locations correspond to the initial positions of user 1 and 2, due to their movement restriction as HMD users (*i.e.*, only teleportation in fixed locations was allowed). More generally, the initial positions of all users are clearly visible in the heatmap; additional yellow spots outside of the predefined circles indicate most

likely regions to be visited by desktop users. Whereas movement is more spread in the living room, in the other smaller room of the house, kitchen and bedroom, it appears much more spatially focused, indicating that participants were more static in these spaces.

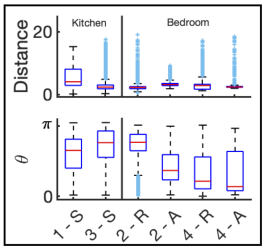
Figure 9.3 displays the boxplot comparison between percentage of motion exhibited by each user. For every frame, we considered the user to be "in motion" if either their relative position with respect to the previous frame changed more than 0.05 cm, or if any of their rotation angles varied by more than 0.01 rad ($0.573°$). Both measures were taken into account to cover both spatial exploration behaviour, as well as changes in viewing angles. The percentage was then computed with respect to the total number of frames. It can be observed that desktop users (user 3 and 4) exhibit a larger percentage of motion over the course of the movie, with respect to HMD users. Motion was present in the first and last chapter with wider distributions, indicating larger variance in the way users behaved, whereas in the second chapter, users generally showed smaller variance in percentage of motion.

Movements analysis: users vs. avatars. We now compare user behaviour with respect to movements of avatars, namely Sarge, Rachel and Armova. To give an idea about avatar displacements and actions in the movie, Figure 9.2 shows distance covered by each character over time with respect to the previous position of the character. We can notice that during *task* phases, no movement is observed as avatars mainly waits for users to make the action. Also, a spike in the distance is observed around minute 5 and this is because Armova and Rachel leave the scene. Finally, Armova does not appear in the scene only in Chapter 1 and 2.

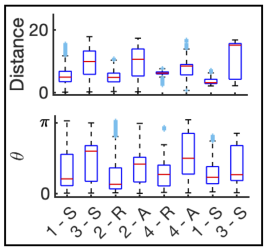
Equipped with this background information, we now study the distribution of relative position and orientation of the users with respect to the avatars, separately per each chapter of the movie (Figure 9.4). In details, the first line of each subplot depicts the distance between each user and each avatar. The second line of subplots shows instead the angle between user's viewing direction and the vector which connects user and avatar at any given time. This angle indicates therefore if user is looking towards direction of character location ($\theta \rightarrow 0$) or in in the opposite direction ($\theta \rightarrow \pi$). In Chapter 1 (Figure 9.4 (a)), it is therefore interesting to notice the different user behaviour between *narrative* and *task* parts of the story. During the three narrative parts, users tend to further explore the environment around them: even if on average, users are looking in the direction of the three virtual characters, their variance indicates a non-uniform behaviour over time, suggesting that participants where also looking around. In the two task parts, instead, the distribution of both spatial distance and angle values is narrower. In particular, for the first task, a wider distribution in terms of viewing angle, which skews further away



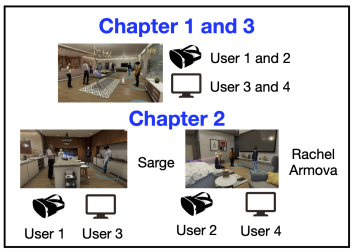
(a) Chapter 1



(b) Chapter 2



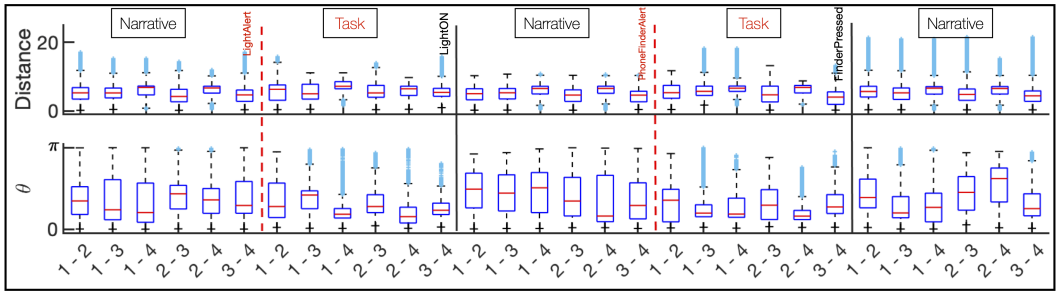
(c) Chapter 3



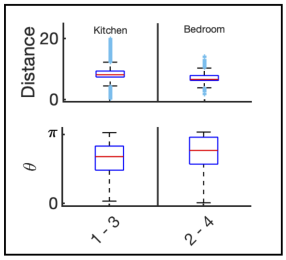
(d) Legend

Figure 9.4: Distribution of spatial distances (first line of each subplot) and angles (second line of each subplot) between users and avatars per each chapter of the story-telling.

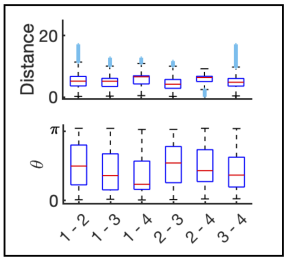
from the avatar, can be observed for user 1, which is the one tasked with pressing the button to turn on the light. This difference might be due to the mismatching of difficulty between the two task. Indeed, participants took on average around 13.33 seconds to switch on the light, while picking up the phone finder controller and press the button required around 25.58 seconds. To validate our intuition, we perform a non-parametric Mann-Whitney statistical test between the distance and angle to avatar recorded during the narrative parts of Chapter 1, and the task parts. A non-parametric test was selected due to the non-normality of the data distribution, according to a Kolmogorov-Smirnoff test. To avoid bias induced by the large number of samples, we performed random sampling on the data, selecting N = 200 samples across the distance vector, and repeating the procedure across 200 sampling runs. We used Fisher's method [233, 234] to combine the probabilities, obtaining that the type of task has a significant effect on the distance ($\chi^2 = 3202.5$, $p < .001$) and on the viewing angle ($\chi^2 = 4771.8$, $p < .001$). In fact, distance to avatar appears to be statistically different between all sub-parts of Chapter 1, indicating varying behavior in terms of spatial movements between as the time progressed. In terms of viewing angle, however, no discernible effect is observed on different narrative parts with respect to the viewing angle (narr.1 - narr.2: $\chi^2 = 407.5$, $p = 0.387$; narr.1 - narr.3: $\chi^2 = 407.5$, $p = 0.388$; narr.2 - narr.3: $\chi^2 = 421.5$, $p = 0.221$), whereas the two tasks exhibited significantly different viewing angle distributions ($\chi^2 = 1270.6$, $p < .001$). In Chapter 2 (Figure 9.4 (b)) participants are moved in different rooms, kitchen and bedroom, both of them smaller compare



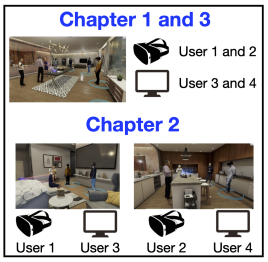
(a) Chapter 1



(b) Chapter 2



(c) Chapter 3



(d) Legend

Figure 9.5: Distribution of spatial distances (first line of each subplot) and angles (second line of each subplot) between couples of users per each chapter of the story-telling.

to the initial living room. This different ambient dimension affects indeed user behaviour: participants in general are much more static compare to the previous chapter. As last observation, we notice that users in Chapter 3 (Figure 9.4 (c)) behave similar to narrative moments of Chapter 1: there are exploration movements both in terms of spatial and angle values.

Movements analysis: users vs. users. Finally, we analyse the user's position and viewing direction with the respect to other participants. As in the previous analysis, Figure 9.5 depicts the spatial distance and angular difference between each couple of users per each chapter of the movie. While in the previous comparison between user and avatars, the behaviour of the latter was known and stayed constant for each experiment, in this case both users under exam have varying positions and viewing directions over time. However, some general behaviour can be extracted also under these conditions confirming the previous findings. For instance, during task parts in Chapter 1 (Figure 9.5 (a)), the distribution of angle values is quite narrow for most of the couples, indicating that they moved their attention from the avatars to another participant. On the contrary, during narrative parts, these distributions are wider highlighting that users were not fixating on each other, rather exploring the scene and the virtual characters within. Statistical tests show that the effect of the task is significant on the viewing angle ($\chi^2 = 2465.6$, $p < .001$), and differences among narrative and task sub-parts of Chapter 1 are always significant

($p < .001$), with the exception of the first and the last part of Chapter 1 ($\chi^2 = 415.9$, $p = 0.282$). In terms of spatial displacements, the distance between users remains low in all the chapters of the movie. The effect of task versus narrative is significant ($\chi^2 = 1270.4$, $p < .001$), and distance always differs significantly between sub-parts ($p < .001$), with the exception of the first two narrative parts ($\chi^2 = 418.8$, $p = 0.249$).

In summary, the following observations can be deduced by the behavioural analysis carried out in this work:

- **Observation 1:** during narrative moments of the story, participants are more inclined to explore the virtual environment with general attention to virtual characters;
- **Observation 2:** the request of interactions with the content by a specific user (*e.g.*, to press a button) leads to reduced movement, while the attention is more focused on the task or on other participants;
- **Observation 3:** the size of the virtual environment in which is located the experience also affect the user's behaviour. In particular, large rooms seem to be more conductive of exploratory behaviour, whereas in smaller rooms, less variation in position or viewing angle is observed.

9.4 Chapter Summary

In this chapter, we analysed the user behaviour during a photorealistic telepresence experiment developed on a volumetric social VR system. We mainly investigated through objective metrics the influence of narrative elements of the story, such as dialogues or interactive task, on participants' movements. Our results show indeed that the motion during the VR experience was affected by the storytelling. More static and focused behaviour happened when a task (either to switch on the light or press a button in a controller) was requested to be done by a specific participant. On the contrary, exploration movements were more frequent when virtual characters were talking in the scene. These observations are key factors to be further investigated, in order to design social VR experiences that can effectively be optimised around the users.

Part V

SUMMARY AND OUTLOOK

Chapter 10

Conclusion and Future Work

10.1 Conclusions

Making remote communications more interactive and immersive is currently a compelling need for the ultimate goal of increasing the quality of life, reducing costs, decreasing our carbon footprint, improving accessibility and equality, and overcoming social difficulties during natural emergencies (*e.g.*, pandemic, tornadoes, etc.). In this context, the research described in the thesis proposed and experimentally demonstrated how to detect key behavioural aspect of people interacting with immersive content; to study the impact of the user behaviour on the system design in novel user-centric solutions. These studies have been first conducted considering navigation only in 3-DoF environments to finally be extended to more challenging systems, such as 6-DoF.

The initial research question that we addressed in this thesis is "**Can we analyse the user behaviour in a 3-DoF system? How?**". We first focused on better understanding and enabling the behavioural analysis of users navigating in a 3-DoF VR environment. Given the importance of this topic, we started answering the question with a depth-in summary of research advances in ODV adaptive streaming, clearly distinguishing works in terms of *system-centric* and *user-centric* streaming solutions (Chapters 2 and 3). System-centric approaches come from a quite straightforward extension of well-established solutions for the 2D video pipeline, adding value in the streaming strategy making it user-aware (*i.e.*, tile-based viewport-dependent streaming). Due to the key role of the users, behavioural investigations on how viewers navigate within 360° video have attracted a lot of interest, showing the benefit of understanding users behaviour, and enabling personalised ODV streaming solutions (*i.e.*, user-centric streaming system). We thus contributed to the overall open problem of behavioural analysis in 3-DoF VR systems by proposing and developing new behavioural analysis tools and methodologies, specifically built

for immersive environments. In Chapter 4, a novel *graph-based method* to identify clusters of users who are attending the same portion of the spherical content over time was defined and tested. The proposed solution takes into account the spherical geometry of the content and aims at clustering users based on the actual overlap of displayed content among users. An extensive behavioural data analysis based on the aforementioned clustering algorithm across content and different VR devices was carried out on a new dataset collected in collaboration with Trinity College Dublin. The analysis revealed useful insights on the effect of both device and content on the navigation. Those could be precious considerations from the system design perspective. We concluded this part of the thesis proposing a different methodology to analyse the behaviour in a 3-DoF VR scenario aimed at characterising navigation patterns not only across content but also across users (Chapter 5). Namely, this chapter presents an *intra-user behavioural analysis* focused on understanding the behaviour of each individual when navigating in VR, and an *inter-user behavioural analysis* aimed at understanding how much information about a single content can be extracted when observing an entire population of viewers. By leveraging on the knowledge from different disciplines, these behavioural investigations are based on information-theoretic metrics. The key intuition is to show that these metrics allow us to quantify the actual behaviour of user's navigation.

The middle part of this thesis is instead aimed at understanding how taking into account the user behaviour can help in enabling novel user-centric solutions. Specifically, we addressed the following research question: "**Does the user behaviour affect the system design? How?**" We presented two case study: an *user-centric optimisation for coding and storage* at the server (Chapter 6) and a *navigation-aware delivery strategy* (Chapter 7) for 3-DoF system. The first was an initial attempt of investigation of behavioural influence on the system design. In particular, we evaluated the optimal set of coding parameters to store ODVs at the main server minimising the total cost and maximising user's experience, taking into account the users behaviour and network characteristics. Results showed that our solution performs well in terms of total cost (*i.e.*, encoding and storage cost) and quality experienced by users. Most importantly, we also highlighted how the different types of user navigation (*e.g.*, affinity) impact on the optimal set. We also proposed an optimal transmission strategy at the client side (*i.e.*, adaptation logic) in a 3-DoF VR system. The novelty was in considering both a viewport-quality as metric that reflects the quality perceived by the end-user but also the popularity of each viewport to be displayed through heatmaps. Results showed a visible gain in terms of navigation quality but, above all, a strong dependency of the optimal

solution on both the video content and user navigation. These two case studies opened the gate to user-centric studies focused on making the users behaviour the driver of 3-DoF VR system designs.

The last part of this thesis moved the focus of attention to a more challenging environment, such as 6-DoF VR systems and was aimed at answering to the following question: "**Can we extend behavioural tools for 3-DoF to 6-DoF system?**" To address this question, in Chapter 7, we first investigated how new physical settings and locomotion functionalities given to users can affect the analysis and the understanding of their behaviour. In particular, we highlighted the main limitations of existing 3-DoF tools when applied to 6-DoF environment. Given these observations, we stated the need for developing new solutions for the analysis of 6-DoF trajectories and thus, we defined *new metrics aimed at capturing users trajectory similarity in 6-DoF*. We validated and tested these metrics on real navigation trajectory collected in a 6-DoF VR environment. Our results showed that solutions that consider both user position on the virtual floor and viewing direction are promising to correctly detect users with a similar behaviour while experiencing volumetric content. We also demonstrated the robustness and versatility of these metrics showing that they preserve good performance on navigation trajectories collected in a 6-DoF Augmented Reality (AR) scenario. Even if users in AR are enabled with the same 6-DoF locomotion settings of VR, the viewing device (*i.e.*, HMD vs. smartphones) and the FoV are different. Despite these differences, our proposed similarity metrics are still good at identifying viewers who are displaying similar content. To the best of our knowledge, at the time of publication, this was the first study proposing behavioural metric specific for 6-DoF environments (both VR and AR settings). We further investigated navigation in a 6-DoF system with a case study in Chapter 9. Here, a *behavioural analysis of user navigating in a 6-DoF social VR movie* was conducted. We mainly investigated how users interactivity is affected by salient agents (*i.e.*, virtual characters) and by narrative elements of the VR movie (*i.e.*, dialogues versus interactive part). Results showed indeed that the motion during the immersive experience was influenced by the storytelling, opening the gate to innovative behavioural investigations to help video directors drive users attention, and thus ensure create a feeling of immersion.

10.2 Future work

The research and discussions presented in this thesis motivate further directions of investigation, some of these are listed below as potential future research lines:

- **Future head motion prediction in a 3-DoF environment:** Despite the in-

tense efforts, viewport prediction research is still under intense investigation and is still suffering from major shortcomings. As described in Chapter 3, some of the existing approaches to predict users trajectories are based on clustering techniques. In particular, Nasrabadi *et al.* propose in [140] a novel viewport prediction algorithm based on our proposed graph-based clustering presented in Chapter 4 of this thesis. Results carried out in their research show good performance in a prediction window of 5 seconds even if highly affected by the video content. Therefore, a promising future direction would be to improve our graph-based clustering to better study predictive mechanisms and reach higher performance. Moreover, novel approaches are now emerging to improve the prediction accuracy, considering different information, such as spatial audio and user emotion. For instance, authors in [235] propose to improve prediction incorporating spatial audio characteristics of the video content. Regarding emotion, there are efforts to enable labelling emoting during VR presentation [236] to create ground truth for user emotion during immersion in ODV. However, a trained model with such data can automatically detect emotion from recorded pupils to perform predictions [237]. Therefore, this introduce a new potential for our proposed graph-based clustering to enhance future predictive algorithm where behavioural information are embedded into spatial audio and user emotion information.

- **User profiling and outliers detection:** A user profile is a collection of information that describes the behavioural features of a user and that is used to identify key behaviours. In immersive context, users profiles can be utilised for different purposes such as enabling new modalities for *viewport prediction*, *live streaming services optimised for users profiles* but also for *user-based QoE assessment methods*. In such cases, no prior users have watched the content, hence any prediction model needs to be trained without an existing dataset of users that experienced such content. In other words, this scenario may require updating the current prediction methods to more online versions. In this context, information extrapolated from users profiles can help overcoming the well known cold start problem. Our proposed information-theoretic analysis presented in Chapter 5 would be beneficial in profiling individually viewers in real-time applications since they can extract behavioural information at very low computational complexity. Another application for user profiling would be to improve quality assessment for immersive content. As emerged in this preliminary work [158], the interactive navigation as well as the fact that users display only a restricted portion of the content

affect the characterisation (*e.g.*, to quantify spatial and temporal complexity, depth features etc.) of ODV. Being able to properly evaluate the complexity of the visual content to be used during subjective test and for benchmarking quality metrics is fundamental to enable comparative studies. Finally, our information-theoretic analysis could be exploited and be also beneficial for detecting outlier trajectories, *i.e.*, users who show different behaviour from others. Outlier detection can in fact improve the accuracy of prediction algorithm: users who show their outlier behaviour are unlike to be useful during training of behavioural information representative of the entire population. Beyond immersive video streaming system applications, being able to detect users who are interacting in a different way from the majority of the population, might be essential for medical purposes such as studying and being treatment mental disorder [129].

- **Behavioural analysis and motion prediction in 6-DoF environment:** The envisioned direction for future multimedia applications is a digital world with volumetric video as media content and where users are allowed to freely and naturally move within the scene at 6-DoF. However, this high level of immersiveness and realism comes with many new open challenges. While content creation and visualisation are relatively well understood, other elements of the pipeline still require research. Despite the progress made in analysing user behaviour in systems with less degree of freedom during the navigation (*e.g.*, 3-DoF), understanding users behaviour in XR/6-DoF spaces is usually overlooked in the literature. The last part of this thesis has already showed the possibility to efficiently extend existing behavioural tools used in 3-DoF to 6-DoF systems. This opens the gate to further investigations aimed at detecting users behavioural differences in a 6-DoF experience, for both VR and AR applications. These tools and investigation are indeed essential to be exploited in efficient user-centric solutions for immersive systems.

Bibliography

[1] R. Trestian, I.-S. Comsa, and M. F. Tuysuz. Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet? *IEEE Communications Surveys & Tutorials*, 20(2):945–977, 2018. doi:10.1109/COMST.2018.2789722.

[2] A. Covaci, R. Trestian, E. B. Saleme, I.-S. Comsa, G. Assres, C. A. Santos, and G. Ghinea. 360° Mulsemedia: A Way to Improve Subjective QoE in 360° Videos. In *Proceedings of ACM International Conference on Multimedia*, pages 2378–2386, 2019. doi:10.1145/3343031.3350954.

[3] J. G. Apostolopoulos, P. A. Chou, B. Culbertson, T. Kalker, M. D. Trott, and S. Wee. The road to immersive communication. *Proceedings of the IEEE*, 100(4):974–990, 2012. doi:10.1109/JPROC.2011.2182069.

[4] Research and markets. Extended Reality (XR) Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). https://www.researchandmarkets.com/reports/4833338/extended-reality-xr-market-growth-trends, 2021. [Online].

[5] C. Flavián, S. Ibáñez-Sánchez, and C. Orús. The impact of virtual, augmented and mixed reality technologies on the customer experience. *Journal of Business Research*, 100:547–560, 2019. doi:10.1016/j.jbusres.2018.10.050.

[6] M. Slater and M. V. Sanchez-Vives. Enhancing our lives with immersive virtual reality. *Frontiers in Robotics and AI*, 3:74, 2016. doi:10.3389/frobt.2016.00074.

[7] M. V. Sanchez-Vives and M. Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, 2005. doi:https://doi.org/10.1038/nrn1651.

[8] M.-L. Ryan. Immersion vs. interactivity: Virtual reality and literary theory. *SubStance*, 28(2):110–137, 1999. doi:10.1353/sub.1999.0015.

[9] J. Mütterlein. The three pillars of virtual reality? investigating the roles of immersion, presence, and interactivity. In *Proceedings of the Hawaii international conference on system sciences*, 2018. doi:10.24251/HICSS.2018.174.

[10] A. Perkis, C. Timmerer, S. Baraković, J. B. Husić, S. Bech, S. Bosse, J. Botev, K. Brunnström, L. Cruz, K. De Moor, et al. QUALINET white paper on definitions of immersive media experience. *European Network on Quality of Experience in Multimedia Systems and Services, 14th QUALINET meeting (online)*, 2020.

[11] J. L. Rubio-Tamayo, M. Gertrudix Barrio, and F. García García. Immersive environments and virtual reality: Systematic review and advances in communication, interaction and simulation. *Multimodal Technologies and Interaction*, 1(4):21, 2017. doi:10.3390/mti1040021.

[12] J. Jerald. *The VR book: Human-centered design for virtual reality*, chapter 2: A history of VR, pages 15–28. Morgan & Claypool, 2015.

[13] M. L. Heilig. Stereoscopic-television apparatus for individual use, Oct. 4 1960. US Patent 2,955,156.

[14] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764, 1968. doi:10.1145/1476589.1476686.

[15] B. Marr. Future predictions of how virtual reality and augmented reality will reshape our lives. www.forbes.com/sites/bernardmarr/2021/06/04/future-predictions-of-how-virtual-reality-and-augmented-reality-will-reshape-our-lives/?sh=4c69a4c868b4, Jun 2021. [Online; last access Jan. 2022].

[16] Facebook. Facebook Horizon— Explore. Play. Create. Together. https://www.oculus.com/facebook-horizon/. [Online; last access Jan. 2022].

[17] Ikon. Interactive museum of monte san michele. https://www.ikon.it/en/projects/mountain-san-michele-museum, Nov 18 2018. [Online; last access Jan. 2022].

[18] Nintendo. Nintendo Labo™ Toy-Con 04: VR Kit. https://www.nintendo.com/products/detail/labo-vr-kit/. [Online; last access Jan. 2022].

[19] Sony. PlayStation VR — Immerse yourself in incredible virtual reality games and experiences. https://www.playstation.com/en-gb/ps-vr/. [Online; last access Jan. 2022].

[20] Game changing digital innovations bring uk museums to new audiences. https://www.ukri.org/news/ukri-and-ahrc-investment-will-help-shape-the-future-of-uk-museums/?utm_medium=email&utm_source=govdelivery, July 2021. [Online; last access Jan. 2022].

[21] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila. A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects. *IEEE Communications Surveys & Tutorials*, 23(2):1160–1192, 2021. doi:10.1109/COMST.2021.3061981.

[22] Nintendo. Pokémon GO. https://pokemongolive.com/en/. [Online; last access Jan. 2022].

[23] P. A. Rauschnabel, A. Rossmann, and M. C. tom Dieck. An adoption framework for mobile augmented reality games: The case of pokémon go. *Computers in Human Behavior*, 76:276–286, 2017. doi:10.1016/j.chb.2017.07.030.

[24] IKEA. IKEA: mobile apps. https://www.ikea.com/gb/en/customer-service/mobile-apps/. [Online; last access Jan. 2022].

[25] Microsoft. What is mixed reality? https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality, August 26 2020. [Online; last access Jan. 2022].

[26] Microsoft. Microsoft HoloLens 2. https://www.microsoft.com/en-us/hololens, 2019. [Online; last access Jan. 2022].

[27] Microsoft. Overview of Dynamics 365 Remote Assist. https://docs.microsoft.com/en-gb/dynamics365/mixed-reality/remote-assist/ra-overview. [Online; last access Jan. 2022].

[28] Microsoft. Types of mixed reality apps. https://docs.microsoft.com/en-us/windows/mixed-reality/design/types-of-mixed-reality-apps, March 21 2018. [Online; last access Jan. 2022].

[29] G. V. Research. Virtual reality market share & trends report, 2021-2028. www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market, March 2021. [Online; last access Jan. 2022].

[30] J. Harth, A. Hofmann, M. Karst, D. Kempf, A. Ostertag, I. Przemus, and B. Schaefermeyer. Different types of users, different types of immersion: A user study of interaction design and immersion in consumer virtual reality. *IEEE Consumer Electronics Magazine*, 7(4):36–43, 2018. doi:10.1109/MCE.2018.2816218.

[31] S. Rossi, A. Guedes, and L. Toni. *Coding, Streaming, and User Behaviour in Omnidirectional Video*. Elsevier, 2022.

[32] S. Rossi, F. De Simone, P. Frossard, and L. Toni. Spherical clustering of users navigating 360° content. In *International Conference on Acoustics, Speech and Signal Processing*, pages 4020–4024. IEEE, 2019. doi:10.1109/ICASSP.2019.8683854.

[33] S. Rossi and L. Toni. Understanding user navigation in immersive experience: an information-theoretic analysis. In *Proceedings of the 12th International Workshop on Immersive Mixed and Virtual Environment Systems*, pages 19–24. ACM, 2020. doi:10.1145/3386293.3397115.

[34] 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*, 16(2):1–26, 2020. https://github.com/V-Sense/VR_user_behaviour; doi:10.1145/3381846.

[35] S. Rossi and L. Toni. Navigation-aware adaptive streaming strategies for omnidirectional video. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2017. doi:10.1109/MMSP.2017.8122230.

[36] S. Rossi, I. Viola, L. Toni, and P. Cesar. A new challenge: Behavioural analysis of 6-DoF user when consuming immersive media. In *International Conference on Image Processing*, pages 3423–3427. IEEE, 2021. doi:10.1109/ICIP42928.2021.9506525.

[37] S. Rossi, L. Viola, Irene Toni, and P. Cesar. From 3-DoF to 6-DoF: New metrics to Analyse Users Behaviour in Immersive Applications. *IEEE Transactions on Image Proccessing*, 2022. [Under revision (submitted December 2021) doi:https://arxiv.org/abs/2112.09402].

[38] S. Rossi, I. Viola, J. Jansen, S. Subramanyam, L. Toni, and C. Pablo. Influence of Narrative Elements on User Behaviour in Photorealistic Social VR. In *ACM*. Association for Computing Machinery (ACM), 2021. doi:10.1145/3458307.3463371.

[39] Google. Street View's 15 favorite Street Views, 2020. [Online; last access Jan. 2022] https://blog.google/products/maps/street-views-15-favorite-street-views/.

[40] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67, 2011.

[41] Meta. Facebook to Acquire Oculus, 2014. [Online; last access Jan. 2022] https://about.fb.com/news/2014/03/facebook-to-acquire-oculus/.

[42] Google. Open sourcing google cardboard, 2019. [Online; last access Jan. 2022] https://developers.googleblog.com/2019/11/open-sourcing-google-cardboard.html.

[43] VRChat. Vrchat raises $1.2 million seed round. https://medium.com/@vrchat/vrchat-raises-1-2-million-seed-round-efba60ea7340, October 2016. [Online; last access Jan. 2022].

[44] Google. A new way to see and share your world with 360-degree video, 2015. [Online; last access Jan. 2022] https://blog.youtube/news-and-events/a-new-way-to-see-and-share-your-world/.

[45] Meta. Insights from a year of 360 videos on facebook, 2017. [Online; last access Jan. 2022] https://facebook360.fb.com/2017/02/16/insights-from-a-year-of-360-videos-on-facebook/.

[46] ISO Central Secretary. Spatial relationship description, generalized URL parameters and other extensions. Standard Tech. Rep., ISO/IEC 23009-1:2014/Amd 2, International Organization for Standardization, Geneva, CH, 2015.

[47] BBC. Click: How we made BBC's first fully 360-degree show, 2016. [Online; last access Jan. 2022] https://www.bbc.com/news/technology-35752662.

[48] DigitalTV Europe. Arte launches virtual reality TV app, 2016. [Online; last access Jan. 2022] https://www.digitaltveurope.com/2016/01/14/arte-launches-virtual-reality-tv-app/.

[49] A. Kipman. Announcing microsoft hololens development edition open for pre-order, shipping march 30. https://blogs.windows.com/devices/2016/02/29/announcing-microsoft-hololens-development-edition-open-for-pre-order-shipping-march-30/, February 2016. [Online; last access Jan. 2022].

[50] Vimeo. Vimeo 360: A home for immersive storytelling, 2017. [Online; last access Jan. 2022] https://vimeo.com/blog/post/introducing-vimeo-360/.

[51] M. Domański, O. Stankiewicz, K. Wegner, and T. Grajek. Immersive visual media—MPEG-I: 360 video, virtual navigation and beyond. In *International Conference on Systems, Signals and Image Processing*, pages 1–9. IEEE, 2017. doi:10.1109/IWSSIP.2017.7965623.

[52] K. Group. Khronos Releases OpenXR 0.90 Provisional Specification for High-performance Access to AR and VR Platforms and Devices. https://www.khronos.org/news/press/khronos-releases-openxr-0.90-provisional-specification-for-high-performance-access-ar-vr-platforms-and-devices, March 2019. [Online; last access Jan. 2022].

[53] G. Luetzenburg, A. Kroon, and A. A. Bjørk. Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences. *Scientific reports*, 11(1):1–9, 2021.

[54] M. Zuckerberg. Founder's letter, 2021. https://about.fb.com/news/2021/10/founders-letter/, October 2021. [Online; last access Jan. 2022].

[55] J. Roach. Mesh for microsoft teams aims to make collaboration in the 'metaverse' personal and fun. https://news.microsoft.com/innovation-stories/mesh-for-microsoft-teams/, November 2021. [Online; last access Jan. 2022].

[56] J. Le Feuvre and C. Concolato. Tiled-based adaptive streaming using MPEG-DASH. In *Proceedings of the International Conference on Multimedia Systems*, pages 1–3. ACM, 2016. doi:10.1145/2910017.2910641.

[57] H. S. Kim, S. B. Nam, S. G. Choi, C. H. Kim, T. T. K. Sung, and C.-B. Sohn. HLS-based 360 VR using spatial segmented adaptive streaming. In *International Conference on Consumer Electronics*, pages 1–4, 2018. doi:10.1109/ICCE.2018.8326272.

[58] H. Bhinde. Gopro unveils odyssey virtual reality camera ring. https://shop.gopro.com/EMEA/cameras/, 2018.

[59] W. Commons. File:omnidirectional camera 01.jpg — wikimedia, the free media repository. "https://en.wikipedia.org/wiki/Omnidirectional_(360-degree)_camera", 2018. [Online; last access Jan. 2022].

[60] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006. doi:10.1561/0600000009].

[61] D. Pio and E. Kuzyakov. Under the hood: Building 360 video, 2015. [Online; last access Jan. 2022] https://engineering.fb.com/2015/10/15/video-engineering/under-the-hood-building-360-video/.

[62] E. Kuzyakov and D. Pio. Next-generation video encoding techniques for 360 video and vr. https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/, Jan 21, 2016. [Online; last access Jan. 2022].

[63] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. doi:https://doi.org/10.1109/TCSVT.2012.2221191.

[64] M. Graf, C. Timmerer, and C. Mueller. Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation. In *Proceedings of ACM Multimedia Systems Conference*, pages 261–271. ACM, 2017. doi:https://doi.org/10.1145/3083187.3084016.

[65] I. Reimat, E. Alexiou, J. Jansen, I. Viola, S. Subramanyam, and P. Cesar. CWIPC-SXR: Point Cloud Dynamic Human Dataset for Social XR. In *Proceedings of ACM Multimedia Systems Conference*, MMSys '21, page 300–306, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3458305.3478452.

[66] R. Pagés, E. Zerman, K. Amplianitis, J. Ondřej, and A. Smolic. Volograms & V-SENSE Volumetric Video Dataset. *ISO/IEC JTC1/SC29/WG07 MPEG2021/m56767*, 2021.

[67] J. van der Hooft, M. T. Vega, T. Wauters, C. Timmerer, A. C. Begen, F. De Turck, and R. Schatz. From capturing to rendering: Volumetric media delivery with six degrees of freedom. *IEEE Communications Magazine*, 58(10):49–55, 2020. doi:10.1109/MCOM.001.2000242.

[68] A. Maglo, G. Lavoué, F. Dupont, and C. Hudelot. 3d mesh compression: Survey, comparisons, and emerging trends. *ACM Computing Surveys*, 47(3):1–41, 2015. doi:https://doi.org/10.1145/2693443.

[69] E. Zerman, C. Ozcinar, P. Gao, and A. Smolic. Textured mesh vs coloured point cloud: A subjective study for volumetric video compression. In *International Conference on Quality of Multimedia Experience*, pages 1–6. IEEE, 2020. doi:10.1109/QoMEX48832.2020.9123137.

[70] J. Jansen, S. Subramanyam, R. Bouqueau, G. Cernigliaro, M. M. Cabré, F. Pérez, and P. Cesar. A pipeline for multiparty volumetric video conferencing: transmission of point clouds over low latency dash. In *Proceedings of ACM Multimedia Systems Conference*, pages 341–344. ACM, 2020. doi:10.1145/3339825.3393578.

[71] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, et al. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and SyVivostems*, 9(1):133–148, 2018. doi:10.1109/JETCAS.2018.2885981.

[72] L. Cui, R. Mekuria, M. Preda, and E. S. Jang. Point-cloud compression: Moving picture experts group's new standard in 2020. *IEEE Consumer Electronics Magazine*, 8(4):17–21, 2019. doi:10.1109/MCE.2019.2905483.

[73] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai. An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing*, 9, 2020. doi:10.1017/ATSIP.2020.12.

[74] V. Oculus et al. Oculus rift. http://www.oculusvr.com/rift, 2018. [Online; last access Jan. 2022].

[75] Google Cardboard – Google VR. https://vr.google.com/cardboard/, 2018. [Online; last access Jan. 2022].

[76] S. Afzal, J. Chen, and K. K. Ramakrishnan. Characterization of 360-degree videos. In *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, page 1–6. ACM, 2017. doi:10.1145/3097895.3097896.

[77] P. R. Alface, J.-F. Macq, and N. Verzijp. Interactive omnidirectional video delivery: A bandwidth-effective approach. *Bell Labs Technical Journal*, 16(4):135–147, 2012. doi:10.1002/bltj.20538.

[78] B. Vishwanath, T. Nanjundaswamy, and K. Rose. Rotational motion model for temporal prediction in 360° video coding. In *International Workshop on Multimedia Signal Processing*. IEEE, 2017. doi:0.1109/MMSP.2017.8122231.

[79] N. M. Bidgoli, R. G. d. A. Azevedo, T. Maugey, A. Roumy, and P. Frossard. OSLO: On-the-sphere learning for omnidirectional images and its application to 360-degree image compression. *arXiv preprint arXiv:2107.09179*, 2021. doi:https://arxiv.org/abs/2107.09179.

[80] E. Kuzyakov and D. Pio. Next-generation video encoding techniques for 360° video and VR, 2016. [Online; last access Jan. 2022] https://engineering.fb.com/2016/01/21/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/.

[81] Z. Xu, X. Zhang, K. Zhang, and Z. Guo. Probabilistic viewport adaptive streaming for 360-degree videos. In *International Symposium on Circuits and Systems*, pages 1–5. IEEE, 2018. doi:10.1109/ISCAS.2018.8351404.

[82] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski. Viewport-adaptive navigable 360-degree video delivery. In *International Conference on Communications*, pages 1–7. IEEE, 2017. doi:10.1109/ICC.2017.7996611.

[83] L. Toni, R. Aparicio-Pardo, K. Pires, G. Simon, A. Blanc, and P. Frossard. Optimal selection of adaptive streaming representations. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(2s):43, 2015. doi:10.1145/2700294.

[84] C. Ozcinar, A. De Abreu, S. Knorr, and A. Smolic. Estimation of optimal encoding ladders for tiled 360 VR video in adaptive streaming systems. In *International Symposium on Multimedia*, pages 45–52. IEEE, 2017. doi:https://doi.org/10.1109/ISM.2017.17.

[85] S. Ahsan, A. Hourunranta, I. D. D. Curcio, and E. Aksu. FriSBE: adaptive bit rate streaming of immersive tiled video. In *Proceedings of the 25th Workshop on Packet Video*, pages 28–34. ACM, 2020. doi:10.1145/3386292.3397121.

[86] J. Fu, X. Chen, Z. Zhang, S. Wu, and Z. Chen. 360SRL: A Sequential Reinforcement Learning Approach for ABR Tile-Based 360 Video Streaming. In *International Conference on Multimedia Expo Workshops*, pages 290–295. IEEE, 2019. doi:10.1109/ICME.2019.00058.

[87] 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*, 9(1):217–230, 2019. doi:10.1109/JETCAS.2019.2895096.

[88] J. Son, D. Jang, and E.-S. Ryu. Implementing Motion-Constrained Tile and Viewport Extraction for VR Streaming. In *Proceedings of the 28th SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 61–66. ACM, 2018. doi:10.1145/3210445.3210455.

[89] N. M. Bidgoli, T. Maugey, and A. Roumy. Fine granularity access in interactive compression of 360-degree images based on rate-adaptive channel codes. *IEEE Transactions on Multimedia*, 23:2868–2882, 2020. doi:10.1109/TMM.2020.3017890.

[90] J. Park, P. A. Chou, and J.-N. Hwang. Rate-utility optimized streaming of volumetric media for augmented reality. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):149–162, 2019. doi:10.1109/JETCAS.2019.2898622.

[91] M. Hosseini and C. Timmerer. Dynamic adaptive point cloud streaming. In *Proceedings of Packet Video Workshop*, pages 25–30. ACM, 2018. doi:10.1145/3210424.3210429.

[92] S. Subramanyam, I. Viola, A. Hanjalic, and P. Cesar. User centered adaptive streaming of dynamic point clouds with low complexity tiling. In *Proceedings of ACM International Conference on Multimedia*, pages 3669–3677. ACM, 2020. doi:10.1145/3394171.3413535.

[93] L. He, W. Zhu, K. Zhang, and Y. Xu. View-dependent streaming of dynamic point cloud over hybrid networks. In *Pacific Rim Conference on Multimedia*, pages 50–58. Springer, 2018. doi:10.1007/978-3-030-00776-8_5.

[94] F. Qian, B. Han, J. Pair, and V. Gopalakrishnan. Toward practical volumetric video streaming on commodity smartphones. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pages 135–140, 2019. doi:10.1145/3301293.3302358.

[95] B. Han, Y. Liu, and F. Qian. ViVo: Visibility-aware mobile volumetric video streaming. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020. doi:10.1145/3372224.3380888.

[96] J. Van Der Hooft, T. Wauters, F. De Turck, C. Timmerer, and H. Hellwagner. Towards 6DoF HTTP adaptive streaming through point cloud compression. In *Proceedings of ACM International Conference on Multimedia*, pages 2405–2413, 2019. doi:10.1145/3343031.3350917.

[97] M. T. Vega, J. van der Hooft, J. Heyse, F. De Backere, T. Wauters, F. De Turck, and S. Petrangeli. Exploring New York in 8K: an adaptive tile-based virtual reality video streaming experience. In *Proceedings of ACM Multimedia Systems Conference*, pages 330–333, 2019. doi:10.1145/3304109.3323831.

[98] B. Han. Mobile immersive computing: Research challenges and the road ahead. *IEEE Communications Magazine*, 57(10):112–118, 2019. doi:10.1109/MCOM.001.1800876.

[99] Z. Chen, Y. Li, and Y. Zhang. Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation. *Signal Processing*, 146:66–78, 2018. doi:10.1016/j.sigpro.2018.01.004.

[100] D. He, C. Westphal, and J. J. Garcia-Luna-Aceves. Network Support for AR/VR and Immersive Video Application: A Survey. In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, pages 359–369. Science and Technology Publications, 2018.

[101] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu. A Survey on 360° Video Streaming: Acquisition, Transmission, and Display. *ACM Computing Surveys*, 52(4):1–36, 2019. doi:10.1145/3329119.

[102] M. Zink, R. Sitaraman, and K. Nahrstedt. Scalable 360° Video Stream Delivery: Challenges, Solutions, and Opportunities. *Proceedings of the IEEE*, 107(4):639–650, 2019. doi:10.1109/JPROC.2019.2894817.

[103] R. G. d. A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard. Visual Distortions in 360° Videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2524–2537, 2020. doi:10.1109/TCSVT.2019.2927344.

[104] A. Yaqoob, T. Bi, and G.-M. Muntean. A survey on adaptive 360° video streaming: Solutions, challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 22(4):2801–2838, 2020. doi:10.1109/COMST.2020.3006999.

[105] R. Shafi, W. Shuai, and M. U. Younus. 360° Video Streaming: A Survey of the State of the Art. *Symmetry*, 12(9):1491, 2020. doi:10.3390/sym12091491.

[106] M. Xu, C. Li, S. Zhang, and P. Le Callet. State-of-the-art in 360° video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):5–26, 2020. doi:10.1109/JSTSP.2020.2966864.

[107] J. Ruan and D. Xie. Networked VR: State of the Art, Solutions, and Challenges. *Electronics*, 10(2):166, 2021. doi:10.3390/electronics10020166.

[108] F. Chiariotti. A survey on 360-degree video: Coding, quality of experience and streaming. *Computer Communications*, 177:133–155, 2021. doi:10.1016/j.comcom.2021.06.029.

[109] X. Corbillon, F. De Simone, and G. Simon. 360-degree video head movement dataset. In *Proceedings of ACM Multimedia Systems Conference*, pages 199–204, 2017. [Online; last access Jan. 2022] https://doi.org/10.1145/3193701; doi:10.1145/3083187.3083215.

[110] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu. 360° video viewing dataset in head-mounted virtual reality. In *Proceedings of ACM Multimedia Systems Conference*, pages 211–216, 2017. [Online; last access Jan. 2022] 10.1145/3192927.

[111] C. Wu, Z. Tan, Z. Wang, and S. Yang. A dataset for exploring user behaviors in VR spherical video streaming. In *Proceedings of ACM Multimedia Systems Conference*, pages 193–198. ACM, 2017. [Online; last access Jan. 2022] https://doi.org/10.1145/3192423.

[112] M. Xu, C. Li, Y. Liu, X. Deng, and J. Lu. A subjective visual quality assessment method of panoramic videos. In *IEEE International Conference on Multimedia Expo Workshops*, pages 517–522. IEEE, 2017. [Online; last access Jan. 2022] https://github.com/Archer-Tatsu/head-tracking.

[113] C. Ozcinar and A. Smolic. Visual attention in omnidirectional video for virtual reality applications. In *2018 Tenth international conference on quality of multimedia experience*, pages 1–6. IEEE, 2018. [Online; last access Jan. 2022] https://github.com/cozcinar/omniAttention.

[114] S. Fremerey, A. Singla, K. Meseberg, and A. Raake. AVtrack360: An open dataset and software recording people's head rotations watching 360° videos on an HMD. In *Proceedings of ACM Multimedia Systems Conference*, pages 403–408. ACM, 2018. [Online; last access Jan. 2022] https://github.com/acmmmsys/2018-AVTrack360.

[115] 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*, 41(11):2693–2708, 2018. [Online; last access Jan. 2022] https://github.com/YuhangSong/DHP.

[116] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet. A dataset of head and eye movements for 360° videos. In *Proceedings of ACM Multimedia Systems Conference*, pages 432–437. ACM, 2018. [Online; last access Jan. 2022] https://salient360.ls2n.fr/datasets/, doi:10.1145/3204949.3208139.

[117] Z. Zhang, Y. Xu, J. Yu, and S. Gao. Saliency detection in 360° videos. In *Proceedings of the European conference on computer vision*, pages 488–

503, 2018. [Online; last access Jan. 2022] https://github.com/xuyanyu-shh/Saliency-detection-in-360-video.

[118] 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 Conference on Computer Vision and Pattern Recognition*, pages 5333–5342. IEEE, 2018. [Online; last access Jan. 2022] https://github.com/xuyanyu-shh/VR-EyeTracking, doi:10.1109/CVPR.2018.00559.

[119] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. McMahan, R. Prakash, M. C. Farias, and M. M. Carvalho. A taxonomy and dataset for 360° videos. In *Proceedings of ACM Multimedia Systems Conference*, pages 273–278, 2019. [Online; last access Jan. 2022] https://github.com/acmmmsys/2019-360dataset; doi:10.1145/3304109.3325812.

[120] M. F. R. Rondón, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso. A unified evaluation framework for head motion prediction methods in 360° videos. In *Proceedings of ACM Multimedia Systems Conference*, pages 279–284, 2020. [Online; last access Jan. 2022] https://gitlab.com/miguelfromeror/head-motion-prediction/tree/master; doi:10.1145/3339825.3394934.

[121] A. Dharmasiri, C. Kattadige, V. Zhang, and K. Thilakarathna. Viewport-aware dynamic 360° video segment categorization. In *Proceedings of the 31st Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 114–121. ACM, 2021. [Online; last access Jan. 2022] https://github.com/theamaya/Viewport-Aware-Dynamic-360-Video-Segment-Categorization; doi:10.1145/3458306.3461000.

[122] 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 *International Conference on Big Data*. IEEE, 2016. doi:10.1109/BigData.2016.7840720.

[123] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang. Pano: Optimizing 360° video streaming with a better understanding of quality perception. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 394–407, 2019. doi:10.1145/3341302.3342063.

[124] J. Chakareski, R. Aksu, V. Swaminathan, and M. Zink. Full UHD 360-Degree Video Dataset and Modeling of Rate-Distortion Characteristics and Head Movement Navigation. In *Proceedings of ACM Multimedia Systems Conference*, pages 267–273, 2021. doi:10.1145/3458305.3478447.

[125] F. Duanmu, Y. Mao, S. Liu, S. Srinivasan, and Y. Wang. A subjective study of viewer navigation behaviors when watching 360-degree videos on computers. In *IEEE International Conference on Multimedia Expo Workshops*, pages 1–6, 2018. doi:10.1109/ICME.2018.8486537.

[126] M. V. d. Broeck, F. Kawsar, and J. Schöning. It's all around you: Exploring 360° video viewing experiences on mobile devices. In *Proceedings of ACM International Conference on Multimedia*, pages 762–768, 2017. doi:10.1145/3123266.3123347.

[127] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, and D. Eager. The prefetch aggressiveness tradeoff in 360° video streaming. In *Proceedings of ACM Multimedia Systems Conference*, pages 258–269, 2018. doi:10.1145/3204949.3204970.

[128] K. Srivastava, R. C. Das, and S. Chaudhury. *Virtual reality applications in mental health: Challenges and perspectives*, volume 23. Wolters Kluwer–Medknow Publications, 2014. doi:10.4103/0972-6748.151666.

[129] R. F. Martin, P. Leppink-Shands, M. Tlachac, M. DuBois, C. Conelea, S. Jacob, V. Morellas, T. Morris, and N. Papanikolopoulos. The Use of Immersive Environments for the Early Detection and Treatment of Neuropsychiatric Disorders. *Frontiers in Digital Health*, 2:40, 2021. doi:10.3389/fdgth.2020.576076.

[130] S. Petrangeli, G. Simon, and V. Swaminathan. Trajectory-based viewport prediction for 360-degree virtual reality videos. In *International Conference on Artificial Intelligence and Virtual Reality*, pages 157–160. IEEE, 2018. doi:10.1109/AIVR.2018.00033.

[131] S. Atev, G. Miller, and N. P. Papanikolopoulos. Clustering of vehicle trajectories. *IEEE transactions on intelligent transportation systems*, 11(3):647–657, 2010. doi:10.1109/TITS.2010.2048101.

[132] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[133] L. Xie, X. Zhang, and Z. Guo. CLS: A Cross-user Learning based System for Improving QoE in 360-degree Video Adaptive Streaming. In *Proceedings of ACM International Conference on Multimedia*, page 564–572, 2018. doi:10.1145/3240508.3240556.

[134] T. Xu, B. Han, and F. Qian. Analyzing viewport prediction under different VR interactions. In *Proceedings of the ACM International Conference on Emerging Networking Experiments And Technologies*, pages 165–171, 2019. doi:10.1145/3359989.3365413.

[135] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *Proceedings of ACM Workshop on All Things Cellular: Operations, Applications and Challenges*, pages 1–6, 2016. doi:10.1145/2980055.2980056.

[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 ACM International Conference on Multimedia*, pages 306–314, 2017. doi:10.1145/3123266.3123453.

[137] D. V. Nguyen, H. T. Tran, A. T. Pham, and T. C. Thang. An optimal tile-based approach for viewport-adaptive 360-degree video streaming. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):29–42, 2019. doi:10.1109/JETCAS.2019.2899488.

[138] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash. Adaptive 360-degree video streaming using scalable video coding. In *Proceedings of ACM International Conference on Multimedia*, pages 1689–1697, 2017. doi:10.1145/3123266.3123414.

[139] 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 *IEEE International Conference on Multimedia Expo Workshops*, pages 1–6, 2018. doi:10.1109/ICME.2018.8486606.

[140] A. T. Nasrabadi, A. Samiei, and R. Prakash. Viewport prediction for 360° videos: a clustering approach. In *Proceedings of ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 34–39, 2020. doi:10.1145/3386290.3396934.

[141] M. Hu, J. Chen, D. Wu, Y. Zhou, Y. Wang, and H.-N. Dai. TVG-Streaming: Learning User Behaviors for QoE-Optimized 360-Degree Video Streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):4107–4120, 2020. doi:10.1109/TCSVT.2020.3046242.

[142] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo. 360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming. In *Proceedings of ACM International Conference on Multimedia*, pages 315–323, 2017. doi:10.1145/3123266.3123291.

[143] F.-Y. Chao, C. Ozcinar, and A. Smolic. Transformer-based Long-Term Viewport Prediction in 360° Video: Scanpath is All You Need. In *International Workshop on Multimedia Signal Processing*. IEEE, 2021.

[144] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of ACL Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020. doi:10.18653/v1/2020.emnlp-demos.6.

[145] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov, and A. Kondoz. Predicting head trajectories in 360° virtual reality videos. In *International Conference on 3D Immersion*, pages 1–6. IEEE, 2017. doi:10.1109/IC3D.2017.8251913.

[146] 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 ACM International Conference on Multimedia*, pages 1190–1198, 2018. doi:10.1145/3240508.3240669.

[147] M. F. R. Rondon, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso. TRACK: A New Method from a Re-examination of Deep Architectures for Head Motion Prediction in 360-degree Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. doi:10.1109/TPAMI.2021.3070520.

[148] X. Zhang, G. Cheung, Y. Zhao, P. Le Callet, C. Lin, and J. Z. G. Tan. Graph learning based head movement prediction for interactive 360 video streaming. *IEEE Transactions on Image Processing*, 30:4622–4636, 2021. doi:10.1109/TIP.2021.3073283.

[149] 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 vir-

tual reality. In *Proceedings of ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 67–72, 2017. doi:10.1145/3083165.3083180.

[150] X. Feng, V. Swaminathan, and S. Wei. Viewport prediction for live 360-degree mobile video streaming using user-content hybrid motion tracking. In *Proceedings of the ACM Interactive, Mobile, Wearable and Ubiquitous Technologies*, volume 3, pages 1–22, 2019. doi:10.1145/3328914.

[151] L. Chopra, S. Chakraborty, A. Mondal, and S. Chakraborty. PARIMA: Viewport Adaptive 360-Degree Video Streaming. In *Proceedings of ACM Web Conference*, pages 2379–2391, 2021. doi:10.1145/3442381.3450070.

[152] J. Park, M. Wu, K.-Y. Lee, B. Chen, K. Nahrstedt, M. Zink, and R. Sitaraman. SEAWARE: Semantic Aware View Prediction System for 360-degree Video Streaming. In *International Symposium on Multimedia*, pages 57–64. IEEE, 2020. doi:10.1109/ISM.2020.00016.

[153] S. Park, M. Hoai, A. Bhattacharya, and S. R. Das. Adaptive streaming of 360-degree videos with reinforcement learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1839–1848, 2021. doi:10.1109/WACV48630.2021.00188.

[154] A. Farhadi and J. Redmon. Yolov3: An incremental improvement. In *Computer Vision and Pattern Recognition*, pages 1804–2767. Springer Berlin/Heidelberg, Germany, 2018.

[155] A. Serrano, V. Sitzmann, J. Ruiz-Borau, G. Wetzstein, D. Gutierrez, and B. Masia. Movie editing and cognitive event segmentation in virtual reality video. *ACM Transactions on Graphics*, 36(4):1–12, 2017. doi:10.1145/3072959.3073668.

[156] M. Yu, H. Lakshman, and B. Girod. A framework to evaluate omnidirectional video coding schemes. In *International Symposium on Mixed and Augmented Reality*, pages 31–36. IEEE, 2015. doi:10.1109/ISMAR.2015.12.

[157] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein. Saliency in VR: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics*, 24(4):1633–1642, 2018. doi:10.1109/TVCG.2018.2793599.

[158] F. De Simone, J. Gutiérrez, and P. Le Callet. Complexity measurement and characterization of 360-degree content. *Electronic Imaging*, Human Vision and Electronic Imaging(12), 2019. doi:10.2352/ISSN.2470-1173.2019.12.HVEI-216.

[159] Y. Rai, P. L. Callet, and P. Guillotel. Which saliency weighting for omni directional image quality assessment? In *International Conference on Quality of Multimedia Experience*, pages 1–6, 2017. doi:10.1109/QoMEX.2017.7965659.

[160] A. Duchowski and G. Marmitt. Modeling Visual Attention in VR: Measuring the Accuracy of Predicted Scanpaths. In *Eurographics - Short Presentations*. Eurographics Association, 2002. doi:10.2312/egs.20021022.

[161] E. Upenik and T. Ebrahimi. A simple method to obtain visual attention data in head mounted virtual reality. In *IEEE International Conference on Multimedia Expo Workshops*, pages 73–78, 2017. doi:10.1109/ICMEW.2017.8026231.

[162] A. Tse, C. Jennett, J. Moore, Z. Watson, J. Rigby, and A. L. Cox. Was I there? Impact of platform and headphones on 360 video immersion. In *Proceedings of the ACM conference on human factors in computing systems*, pages 2967–2974, 2017. doi:10.1145/3027063.3053225.

[163] S. W. Bindman, L. M. Castaneda, M. Scanlon, and A. Cechony. Am I a bunny? The impact of high and low immersion platforms and viewers' perceptions of role on presence, narrative engagement, and empathy during an animated 360 video. In *Proceedings of the ACM conference on human factors in computing systems*, pages 1–11, 2018. doi:10.1145/3173574.3174031.

[164] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9), 1973.

[165] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. doi:10.1088/1742-5468/2008/10/p10008.

[166] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979. doi:https://doi.org/10.2307/2346830.

[167] ITU-T. Subjective video quality assessment methods for multimedia applications. ITU-T Recom. P.910, Apr 2008.

[168] A. D. Abreu, C. Ozcinar, and A. Smolic. Look around you: Saliency maps for omnidirectional images in vr applications. In *International Conference on Quality of Multimedia Experience*, pages 1–6. IEEE, 2017. doi:10.1109/QoMEX.2017.7965634.

[169] M. M. Ramey, A. P. Yonelinas, and J. M. Henderson. Conscious and unconscious memory deferentially impact attention: Eye movements, visual search, and recognition processes. *Cognition*, 185(1):71–82, 2019. doi:10.1016/j.cognition.2019.01.007.

[170] L. C. Loschky, A. M. Larson, J. P. Magliano, and T. J. Smith. What would Jaws do? The tyranny of film and the relationship between gaze and higher-level narrative film comprehension. *PloS one*, 10(11), 2015. doi:10.1371/journal.pone.0142474.

[171] A. Nguyen and Z. Yan. A saliency dataset for 360-degree videos. In *Proceedings of ACM Multimedia Systems Conference*, page 279–284. ACM, 2019. doi:10.1145/3304109.3325820.

[172] A. Cuttone, S. Lehmann, and M. González. Understanding predictability and exploration in human mobility. *EPJ Data Science*, 2018. doi:10.1140/epjds/s13688-017-0129-1.

[173] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010. doi:10.1126/science.1177170.

[174] T. Cover and J. Thomas. *Elements of information theory*. Wiley & Sons, 2012.

[175] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.

[176] N. M. Timme and C. Lapish. A tutorial for information theory in neuroscience. *eNeuro*, 2018. doi:10.1523/ENEURO.0052-18.2018.

[177] D. do Couto Teixeira, J. M. Almeida, and A. C. Viana. On estimating the predictability of human mobility: the role of routine. *EPJ Data Science*, 10(1):49, 2021. doi:10.1140/epjds/s13688-021-00304-8.

[178] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 24(5):530–536, 1978. doi:10.1109/TIT.1978.1055934.

[179] O. Le Meur, T. Baccino, and A. Roumy. Prediction of the inter-observer visual congruency (IOVC) and application to image ranking. In *Proceedings of ACM International Conference on Multimedia*, 2011. doi:10.1145/2072298.2072347.

[180] B. John, P. Raiturkar, O. Le Meur, and E. Jain. A Benchmark of Four Methods for Generating 360 Saliency Maps from Eye Tracking Data. *International Journal of Semantic Computing*, 2019. doi:10.1142/S1793351X19400142.

[181] MathWorks. Motion-based multiple object tracking, 2020. [Online; last access Jan. 2022] www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html.

[182] 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*, 7(6):969–977, 2013. doi:10.1109/JSTSP.2013.2271451.

[183] A. Mahzari, A. Taghavi Nasrabadi, A. Samiei, and R. Prakash. FoV-aware edge caching for adaptive 360° video streaming. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 173–181, 2018. doi:10.1145/3240508.3240680.

[184] P. Maniotis, E. Bourtsoulatze, and N. Thomos. Tile-based joint caching and delivery of 360° videos in heterogeneous networks. *IEEE Transactions on Multimedia*, 22(9):2382–2395, 2019. doi:10.1109/TMM.2019.2957993.

[185] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics*, 35(6):1–12, 2016. doi:10.1145/2980179.2980246.

[186] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim. MPEG DASH SRD: spatial relationship description. In *Proceedings of ACM Multimedia Systems Conference*, page 5, 2016. doi:10.1145/2910017.2910606.

[187] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski. Optimal set of 360-degree videos for viewport-adaptive streaming. In *Proceedings of ACM International Conference on Multimedia*, pages 943–951, 2017. doi:10.1145/3123266.3123372.

[188] C. Ozcinar, A. De Abreu, and A. Smolic. Viewport-aware adaptive 360 video streaming using tiles for virtual reality. In *International Conference on Image Processing*, pages 2174–2178. IEEE, 2017. doi:10.1109/ICIP.2017.8296667.

[189] M. Xiao, C. Zhou, Y. Liu, and S. Chen. OpTile: Toward Optimal Tiling in 360-degree Video Streaming. In *Proceedings of ACM International Conference on Multimedia*, pages 708–716, Mountain View California USA, 2017. doi:10.1145/3123266.3123339.

[190] C. Timmerer. Immersive media delivery: Overview of ongoing standardization activities. *IEEE Communications Standards Magazine*, 1(4):71–74, 2017. doi:10.1109/MCOMSTD.2017.1700038.

[191] J. Zou, C. Li, C. Liu, Q. Yang, H. Xiong, and E. Steinbach. Probabilistic tile visibility-based server-side rate adaptation for adaptive 360-degree video streaming. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):161–176, 2019. doi:10.1109/JSTSP.2019.2956716.

[192] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan. Viewport-driven rate-distortion optimized 360° video streaming. In *International conference on communications*, pages 1–7. IEEE, 2018. doi:10.1109/ICC.2018.8422859.

[193] IBM. ILOG CPLEX optimization studio. https://www-01.ibm.com/software/, 2013.

[194] Y. Sun, A. Lu, and L. Yu. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. *IEEE signal processing letters*, 24(9):1408–1412, 2017. doi:10.1109/LSP.2017.2720693.

[195] Amazon. Elastic transcoder pricing. https://aws.amazon.com/elastictranscoder/pricing/, 2019.

[196] Amazon. Cloud storage pricing. https://aws.amazon.com/s3/pricing/, 2019.

[197] J.-R. Ohm and G. Sullivan. Vision, applications and requirements for high efficiency video coding (HEVC). Technical report, ISO/IEC JTC1/SC29/WG11, 2011.

[198] M. Inc. x265 HEVC Encoder / H.265 Video Codec. http://x265.org/, 2018.

[199] Apple. Hls authoring specification for apple devices. = https://developer.apple.com, 2018.

[200] Netflix. Per-title encode optimization. https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2, 2015.

[201] M. Hosseini and V. Swaminathan. Adaptive 360 VR video streaming: Divide and conquer. In *International Symposium on Multimedia*, pages 107–110. IEEE, 2016. doi:10.1109/ISM.2016.0028.

[202] F. De Simone, P. Frossard, P. Wilkins, N. Birkbeck, and A. Kokaram. Geometry-driven quantization for omnidirectional image coding. In *Picture Coding Symposium*, pages 1–5. IEEE, 2016. doi:10.1109/PCS.2016.7906402.

[203] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012. doi:https://doi.org/10.1109/TCSVT.2012.2221191.

[204] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. *Advances in Neural Information Processing Systems*, 2007.

[205] P. Cipresso, I. A. C. Giglioli, M. A. Raya, and G. Riva. The past, present, and future of virtual and augmented reality research: a network and cluster analysis of the literature. *Frontiers in psychology*, 9:2086, 2018. doi:10.3389/fpsyg.2018.02086.

[206] E. Alexiou, N. Yang, and T. Ebrahimi. PointXR: A toolbox for visualization and subjective evaluation of point clouds in virtual reality. In *International Conference on Quality of Multimedia Experience*, pages 1–6. IEEE, 2020. doi:10.1109/QoMEX48832.2020.9123121.

[207] S. Subramanyam, J. Li, I. Viola, and P. Cesar. Comparing the Quality of Highly Realistic Digital Humans in 3DoF and 6DoF: A Volumetric Video Case Study. In *IEEE Conference on Virtual Reality and 3D User Interfaces*, 2020. doi:10.1109/VR46266.2020.00031.

[208] E. Zerman, R. Kulkarni, and A. Smolic. User behaviour analysis of volumetric video in augmented reality. In *International Conference on Quality of Multimedia Experience*, pages 129–132. IEEE, 2021.

[209] C. Swindells, B. A. Po, I. Hajshirmohammadi, B. Corrie, J. Dill, B. Fisher, and K. Booth. Comparing CAVE, wall, and desktop displays for navigation and wayfinding in complex 3D models. In *IEEE Proceedings Computer Graphics International*, 2004. doi:10.1109/CGI.2004.1309243.

[210] C. Christou, A. Tzanavari, K. Herakleous, and C. Poullis. Navigation in virtual reality: Comparison of gaze-directed and pointing motion control. In *Mediterranean Electrotechnical Conference*, 2016. doi:10.1109/MELCON.2016.7495413.

[211] H. Jun, M. R. Miller, F. Herrera, B. Reeves, and J. N. Bailenson. Stimulus sampling with 360-videos: Examining head movements, arousal, presence, simulator sickness, and preference on a large sample of participants and videos. *IEEE Transactions on Affective Computing*, 2020. doi:10.1109/TAFFC.2020.3004617.

[212] J. Bermejo-Berros and M. A. G. Martínez. The relationships between the exploration of virtual space, its presence and entertainment in virtual reality, 360° and 2d. *Virtual Reality*, pages 1–17, 2021. doi:10.1007/s10055-021-00510-9.

[213] D. D. R. Morais, L. S. Althoff, R. Prakash, M. M. Carvalho, and M. C. Farias. A content-based viewport prediction model. In *Electronic Imaging*, number 9, 2021. doi:10.2352/ISSN.2470-1173.2021.9.IQSP-255.

[214] D. Freeman, S. Reeve, A. Robinson, A. Ehlers, D. Clark, B. Spanlang, and M. Slater. Virtual reality in the assessment, understanding, and treatment of mental health disorders. *Psychological medicine*, 47(14):2393–2400, 2017. doi:10.1017/S003329171700040X.

[215] C. N. Geraets, E. C. van der Stouwe, R. Pot-Kolder, and W. Veling. Advances in immersive virtual reality interventions for mental dis-

orders: a new reality? *Current opinion in psychology*, 41:40–45, 2021. doi:10.1016/j.copsyc.2021.02.004.

[216] E. D. Ragan, S. Scerbo, F. Bacim, and D. A. Bowman. Amplified head rotation in virtual reality and the effects on 3D search, training transfer, and spatial orientation. *IEEE transactions on visualization and computer graphics*, 23(8):1880–1895, 2016. doi:10.1109/TVCG.2016.2601607.

[217] H. Creagh. Cave automatic virtual environment. In *Proceedings: Electrical Insulation and Electrical Manufacturing and Coil Winding Technology Conference*, pages 499–504, 2003. doi:10.1109/EICEMC.2003.1247937.

[218] W. Chen, A. Plancoulaine, N. Férey, D. Touraine, J. Nelson, and P. Bourdot. 6DoF navigation in virtual worlds: comparison of joystick-based and head-controlled paradigms. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 2013. doi:10.1145/2503713.2503754.

[219] M. Krivokuća, P. Chou, and P. Savill. 8i voxelized surface light field (8iVSLF) dataset. In *ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914*, 2018.

[220] L. Stankovic, D. P. Mandic, M. Dakovic, I. Kisil, E. Sejdic, and A. G. Constantinides. Understanding the basis of graph signal processing via an intuitive example-driven approach. *IEEE Signal Processing Magazine*, 36(6):133–145, 2019. doi:10.1109/MSP.2019.2929832.

[221] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[222] Volograms. Volograms homepage. `https://www.volograms.com/`. [Online; last access Jan. 2022].

[223] F. Moustafa and A. Steed. A longitudinal study of small group interaction in social virtual reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 2018. doi:10.1145/3281505.3281527.

[224] J. Li, V. Vinayagamoorthy, R. Schwartz, W. IJsselsteijn, D. A. Shamma, and P. Cesar. Social VR: A New Medium for Remote Communication and Collaboration. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, 2020. doi:10.1145/3411763.3441346.

[225] G. Freeman and D. Maloney. Body, avatar, and me: The presentation and perception of self in social virtual reality. *Proceedings of Human-Computer Interaction*, 2021. doi:10.1145/3432938.

[226] J. McVeigh-Schultz, A. Kolesnichenko, and K. Isbister. Shaping Pro-Social Interaction in VR: An Emerging Design Framework. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, 2019. doi:10.1145/3290605.3300794.

[227] D. A. Le, B. MacIntyre, and J. Outlaw. Enhancing the experience of virtual conferences in social virtual environments. In *Conference on Virtual Reality and 3D User Interfaces*, pages 485–494. IEEE, 2020. doi:10.1109/VRW50115.2020.00101.

[228] J. Williamson, J. Li, V. Vinayagamoorthy, D. A. Shamma, and P. Cesar. Proxemics and social interactions in an instrumented virtual reality workshop. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, 2021. doi:10.1145/3411764.3445729.

[229] M. E. Latoschik, D. Roth, D. Gall, J. Achenbach, T. Waltemate, and M. Botsch. The effect of avatar realism in immersive social virtual realities. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, 2017. doi:10.1145/3139131.3139156.

[230] G. Gamelin, A. Chellali, S. Cheikh, A. Ricca, C. Dumas, and S. Otmane. Point-cloud avatars to improve spatial communication in immersive collaborative virtual environments. *Personal and Ubiquitous Computing*, 2020. doi:10.1007/s00779-020-01431-1.

[231] A. Revilla, S. Zamarvide, I. Lacosta, F. Perez, J. Lajara, B. Kevelham, V. Juillard, B. Rochat, M. Drocco, N. Devaud, et al. A collaborative vr murder mystery using photorealistic user representations. In *Conference on Virtual Reality and 3D User Interfaces*, pages 766–766. IEEE, 2021. doi:10.1109/VRW52623.2021.00266.

[232] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner. G* Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods*, 39(2):175–191, 2007. doi:10.3758/BF03193146.

[233] R. A. Fisher et al. Statistical methods for research workers. *Statistical methods for research workers.*, (5th Ed), 1934.

[234] J. T. Kost and M. P. McDermott. Combining dependent p-values. *Statistics & Probability Letters*, 60(2):183–190, 2002. doi:10.1016/S0167-7152(02)00310-3.

[235] F.-Y. Chao, C. Ozcinar, L. Zhang, W. Hamidouche, O. Deforges, and A. Smolic. Towards Audio-Visual Saliency Prediction for Omnidirectional Video with Spatial Audio. In *International Conference on Visual Communications and Image Processing*, pages 355–358. IEEE, 2020. doi:10.1109/VCIP49819.2020.9301766.

[236] T. Xue, A. E. Ali, T. Zhang, G. Ding, and P. Cesar. RCEA-360VR: Real-time, Continuous Emotion Annotation in 360° VR Videos for Collecting Precise Viewport-dependent Ground Truth Labels. In *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021. doi:10.1145/3411764.3445487.

[237] L. J. Zheng, J. Mountstephens, and J. Teo. Four-class emotion classification in virtual reality using pupillometry. *Journal of Big Data*, 7(1):1–9, 2020. doi:10.1186/s40537-020-00322-9.