M 2016

U. PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

ADAPTAÇÃO AUTOMÁTICA DE VISTAS EM APLICAÇÕES 3D

JOSÉ DIOGO FARIA SILVA COSTA
DISSERTAÇÃO DE MESTRADO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automatic adaptation of views in 3D applications

José Diogo Faria Silva Costa

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Maria Teresa Andrade (PhD)

Co-supervisor: Tiago Soares da Costa (MSc)

February 17, 2016

Resumo

Esta dissertação foi desenvolvida no âmbito do projeto de investigação "Personalized Multiview Environment" realizado no Centro de Telecomunicações e Multimédia do INESC TEC e descreve um sistema de FTV capaz de oferecer uma experiência mutlitvista completa e aplicável a diferentes cenários de VOD.

O sistema garante um aproveitamento eficiente dos recursos da rede, através da utilização de MPEG-DASH. Uma *Web camera* monitoriza os movimentos da cabeça do utilizador e possibilita a exploração em 3D de diversos pontos de vista de uma cena, adaptando dinâmicamente o conteúdo ao foco de atenção do utilizador.

Este sistema oferece garantias de boa qualidade de serviço para diversas condições de rede.

Um algoritmo de adaptação automática da qualidade, ajusta o conteúdo mutlivista às condições da rede, podendo ser adaptado a diversos serviços de *streaming* de vídeo.

O sistema permite aos *content providers* oferecer novos formatos de vídeo, sem sobrecarga nas redes de distribuição.

Abstract

This dissertation was developed under the scope of the "Personalized Multiview Environment" project conducted by the Center for Telecommunications and Multimedia at INESC TEC and describes an FTV system capable of delivering a complete multiview experience, applicable to a different range of VOD streaming scenarios.

Using MPEG-DASH for streaming and a Web camera for head tracking, the system allows the exploration of 3D scenes from different viewpoints by dynamically switching viewpoints to the point of focus of the user, while maintaining an efficient use of network resources.

This system promises a good quality of experience for a wide range of network availability scenarios.

A quality adaptation algorithm dynamically adapts the multiview content to the bandwidth availability and can be adapted to any common video streaming service.

This system allows content providers to effortlessly offer new types of video content, without an extra load on the CDN.

Agradecimentos

Agradeço à minha orientadora, a professora Professora Doutora Maria Teresa Andrade, pelo desafio proporcionado.

Ao meu co-orientador, o engenheiro Tiago Soares da Costa, pela simpatia, apoio, prontidão e disponibilidade.

À Faculdade de Engenharia da Universidade do Porto por me oferecer uma formação de qualidade em temas diversificados e fascinantes e por me ajudar a perceber a importância do esforço e do método de trabalho.

Agradeço ao INESC TEC o estágio académico, certamente terá uma grande influência no meu futuro.

Quero agradecer à minha família por me proporcionar as melhores condições para concretizar a minha formação académica.

Agradeço ao meus amigos o apoio e os momentos de divertimento.

Finalmente, agradeço a todos aqueles que contribuíram ou contribuem para o desenvolvimento da ciência, tecnologia e dos conteúdos mutlimédia, pela democratização da informação, pela inspiração e pelo futuro.

Diogo Costa

"There is something that 3D gives to the picture that takes you into another land and you stay there and it's a good place to be. "

Martin Scorsese

Contents

List of Figures

List of Tables

Symbols and Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
DC	Direct Current
GUI	Graphical user interface
f/s	Frames per second
HD	High-Definition
HMD	Head-Mounted Display
HTTP	Hypertext Transfer Protocol
HVS	Human Visual System
IP	Internet Protocol
MP4	MPEG-4 Part 14
QoE	Quality of Experience
QoS	Quality of service
RTT	Round-Trip Time
SMV	Super Multiview
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UHD	Ultra-high-definition
UI	User interface
UX	User EXperience
URL	Uniform Resource Locator
VOD	Video On Demand
VR	Virtual Reality
XML	EXtensible Markup Language
b/s	bits per second

Chapter 1

Introduction

The objective of this chapter is to provide the context and an overview over the dissertation theme and the structure of this document.

1.1 Motivation

Since the release of the first anaglyph film, "The Power of Love" in 1922, 3D films have experienced short-lived periods of moderate success. From the golden age of 3D in the 1950s to the releases in the 1980s, the resurgence of 3D films has been usually accompanied by some technical advance, but this was never sufficient to catch the interest of mainstream audiences or to find a sustainable market [1].

In 2009, with the release of "Avatar", the highest-grossing movie of all time [2], many considered the paradigm was about to change. After "Avatar", the number of films released in 3D dramatically increased [3], launching a renewed interest in the format. Worldwide, theaters with support for 3D saw a tremendous rise [4], manufacturers launched a broad range of 3D screens, television networks created 3D programs and channels, and support for 3D was introduced in several video games [1].

However, by the end of 2013, with a decreasing number of releases and a declining take-rate, many argued that the interest in 3D was declining and the format would again disappear. Many factors were appointed to the failure of 3D, including the ticket prices, the low quality of the releases, the requirement of glasses and the visual discomfort [5] [6].

With the huge success of 3D films in 2015 [7] and with a steady number of 3D releases already planned until 2020 [8], it appears that the novelty effect wore off, but not the interest in the format. It can be stated that both the studios and the audiences got more selective in the choice of 3D films and became more interested in quality over quantity [9]. This trend seems to follow the demand for high-quality and immersive content shown in the rising interest in HMD [10], 360^{o} videos [11] [12], augmented reality and virtual reality devices [13].

On the other hand, the 3D television market never seemed to have caught the interest of the general public [14]. It can be argued that the 3D video format may be exclusive to the theaters and the future of domestic consumption of 3D may rely on HMDs if the availability of autostereoscopic displays or others displaying alternatives remains scarce.

It is also interesting to note the crescent interest in media streaming and on-demand services [15] [16] [17]. The consumption of content is quickly moving from traditional formats towards the Internet [18] with dozens of services operating around the world and offering colossal numbers of affordable, easy-to-access content [19] [20]. These services are dominating how people consume content and it's safe to assume they will become the new standard for multimedia consumption.

This dissertation is part of a project aiming to build an immersive 3D multiview video on demand streaming system.

The possibility of streaming interactive multiview 3D content may open the door to new Free-viewpoint Television (FTV) applications, from the streaming of sports or cultural events to the exploration of environments, objects or monitoring scenes in education, industrial or security applications. FTV can achieve a level of immersion not possible with any other 3D format, by giving the user freedom and control over his viewpoints.

These possibilities may be the driving force for the establishment of domestic consumption of 3D, granting support for new kinds media and applications beyond films.

In the near future, although HMDs may become the *status quo* for the viewing of interactive and 3D content, autostereoscopic displays appear as the ideal alternative for FTV applications and for the introduction of glass-less and comfortable 3D to the home. Smartphones can also offer great results for single-user experiences, but multi-view systems with support for multiple users are still underdeveloped [21].

These technologies might benefit from the introduction of the UHD displays, but it remains a challenge to establish a device for the displaying of 3D and FTV content, that is sustainable for the manufacturers and comfortable and affordable to the users.

1.2 Problem Description

This dissertation was developed as part of the "Personalized Multiview Environments" research project conducted by the Center for Telecommunications and Multimedia at INESC TEC.

The goal of the project is to create a complete FTV system on a client-server architecture, capable of providing an immersive experience by allowing the user to interactively control the viewpoint of a 3D video scene while accounting for terminal and network limitations.

A prototype was built based on MPEG-DASH and includes an HTTP server, a media player, a client application and a Web camera. The HTTP server contains the media content of multiple 3D views of a single scene, obtained by an array of 2D cameras placed alongside in arch. One of the components of the client, the client application, uses the user's head tracking information to identify and select the view to be displayed and sends this information and other parameters to the media player. The media player is responsible for streaming and displaying the content and to manage all the operations requested by the client. Manual control is also available.

Some features of the prototype are underdeveloped and manifest several problems. As such, the main concern of this dissertation is the identification and resolution of the prototype major problems and the implementation of new features that should take into account matters of streaming quality and bandwidth availability.

The ultimate goal of this dissertation is to finish and perfect the prototype and to achieve a fully-functional FTV system, applicable to a different range of scenarios of VOD streaming. It is mandatory to ensure quality of service is guaranteed for a wide range of network availability scenarios and the best quality of experience is provided in order to deliver a complete and immersive experience. To meet the goal, a deep knowledge, and understanding of all the several fields related to FTV, as well as the tools, frameworks, and working platforms should be acquired.

1.3 Document Structure

This document is divided into eight chapters. Every chapter contains an introductory description of the chapter structure and theme.

Chapter 2 concerns the State of the Art. This chapter provides an up-to-date- overview on the most significant topics related to the dissertation theme.

Chapter 3 describes the methodology, presenting the project objectives and the chosen approach.

Chapter 4 thoroughly describes the prototype, highlighting the difference between the initial and final versions of the prototype and the new developments.

The next two chapters address the dissertation objectives. Chapter 5 addresses the issues related to the quality of service and chapter 6 the issues related to the quality of experience.

Chapter 7 describes the tests performed to the several features of the prototype.

Finally, chapter 8 presents the project results, the dissertation conclusion, and final remarks.

1.4 Website

A website was developed and is available at `https://paginas.fe.up.pt/~ee09033/` containing all the all the necessary information about the dissertation from the project description and objectives to the team and developed work. This website was continuously updated throughout the duration of the dissertation.

1.5 Article

A brief article summarizing all the work done was written according to the faculty requirements and guidelines. This article contains a description of the project, the objectives, the developed work and the conclusions.

Chapter 2

State of the Art

This chapter presents an overview of the most significant topics related to the dissertation theme in order to support a solid theoretical basis and a better understanding of the concepts and technological tools regarding the addressed problem. This literature review starts with an outline of the human visual system, taking into account matters of visual comfort and how the HVS is related to the way 3D viewing technologies are built. It follows with a study of 3D viewing technologies and respective compression standards, mechanisms for adaptation of views in multiview systems, along with new and upcoming technologies that might provide a similar experience to multiview video, such as 360^{o} video. Next, an analysis of MPEG-DASH streaming techniques is performed. Finally, an outline of the main machine learning algorithms for prediction, using metadata, is presented, for reference of future implementations on the prototype.

2.1 Human Visual System

Human perception of depth is made possible primarily by accommodation, convergence and the separation between the left and right eyes.

Accommodation refers to a variation of the focal length that allows the eye to focus on an object, as its distance varies. Convergence refers to the muscular rotation of the eyeballs that allow the eyes to converge to the same object.

The separation between the left and right eyes results in a disparity on the images captured by each eye. This disparity depends on the object distance and angle and is called binocular disparity or binocular parallax.

There are different types of parallax. Zero parallax, when the image is perceived by the viewer, in the *viewer space*, at screen level (i.e., *screen space*). Negative parallax, when the image is perceived between the *viewer space* and the *screen space* and positive parallax when the image is perceived beyond the *screen space*, as depicted in Figure 2.1.

Monocular and binocular depth cues also play and important role in depth perception. The most relevant monocular cues are occlusion, (i.e., overlapping of objects), linear perspective (i.e, parallel lines converge to the horizon), prior knowledge and relative sizes of objects, atmospheric

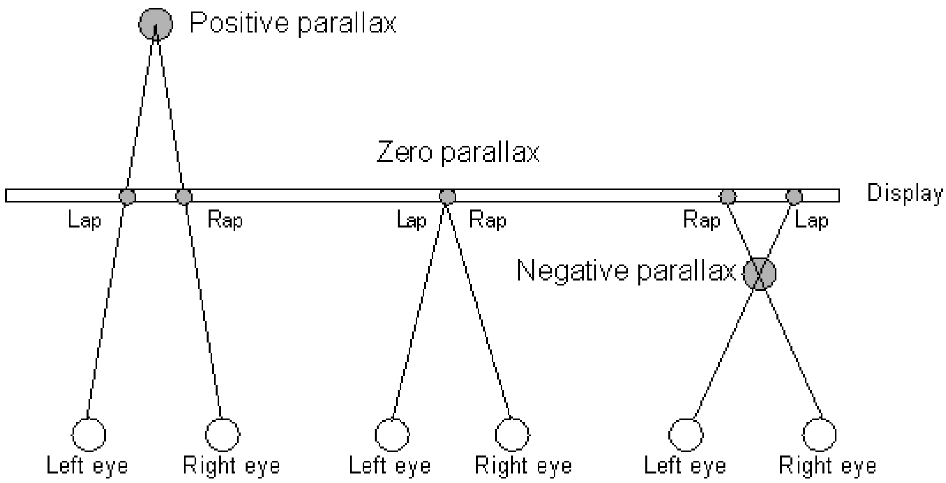

Figure 2.1: Types of parallax
(Source: Digital Media Lab, 2003 [22])

perspective (i.e., loss of contrast and color on distance objects) and selective focus. Light and shade also play an important role in volume, and consequently, depth perception, as does texture.

Among binocular depth cues, the most relevant are the binocular parallax, convergence and motion parallax (i.e., when in motion, nearby objects appear to move to move faster that far away objects).

The importance of depth cues in the depth perception depends primarily on the viewing distance. Occlusion and relative size are independent of distance.

3D viewing technology is built upon the characteristics of depth perception of the HVS and takes advantage of binocular disparity or binocular parallax, motion parallax, convergence, and accommodation. The slightly different signals captured by each eye are combined in the visual cortex, allowing the perception of depth [23] [24] [25].

2.1.1 Visual Comfort

In a recent article, *"Overview of Measurement Methods for Factors Affecting the Human Visual System in 3D Displays"* [26] , M. Park et al. summarize the several discomforting and harmful effects of 3D displays. In sum, unnatural viewing conditions are the main responsible for adverse effects on the observer. Crosstalk, excessive disparity, and accommodation-convergence mismatch are identified as the main sources of visual fatigue. These phenomena are further described ahead.

It is important to address that visual fatigue and discomfort are not exclusive to stereoscopic 3D images and are associated with the intensive use of visual displays. Visual fatigue is also dependent of the susceptibility of the observer, as the ability to perceive depth varies with age and visual capability. Several advancements in the reduction of visual fatigue have been made in different types of technologies (e.g., the super-multiview configuration).

M. Park et al. [26] show that autostereoscopic and multiview systems improve the visual quality and comfort of viewers of 3D content and that the best way to improve the visual comfort is by reproducing natural viewing conditions. They also suggest that holography may be the best way to display 3D content in natural viewing conditions, despite its current technological limitations.

2.1.1.1 Crosstalk

Crosstalk is experienced in autostereoscopic 3D displays where an image from a view intended to one eye is leaked to another. It can also be found in stereoscopic displays if the frequency of the shutter glasses is not well synchronized with the projector. It is recommended to keep the crosstalk level bellow 4% to maintain satisfactory image quality [26] [27].

2.1.1.2 Accommodation-convergence

Accommodation-convergence mismatch occurs when there is a significant disparity between the accommodation and convergence, or if the disparity is too sudden. In a broader description, accommodation refers to the distance to the point of focus (i.e., *screen space*), whilst convergence refers to the level of the object of focus, variable in 3D systems.

Accommodation-convergence can be found in stereoscopic systems. Other effects like the lack of a motion parallax, where a system only displays two sub-views, not taking into account the user position, and the requirement of special eyewear dramatically reduce the user's comfort [26] [27].

2.2 3D Viewing Technologies

The basic principle for displaying 3D video is stereoscopy, a technique for creating or enhancing the illusion of depth, taking advantage of the perception traits of the HVS.

Stereoscopy is usually achieved by displaying offset images separately to the left and to the right eyes of the viewer.

Stereoscopic displaying methods require the use of headgear or special eyewear with filters, for separating the different images to each eye. Other methods, called Autostereoscopic, split the images directly into the viewer's eye, without the requirement of any type of filters or headgear.

The following section lists the most notable 3D displaying technologies.

2.2.1 Stereoscopic Methods

2.2.1.1 Anaglyph 3D

Anaglyph 3D uses passive filters of different colors, typically, red and cyan to encode each eye image.

Anaglyph images are produced by combining the images for the left and right eyes using complementary coding techniques. Typically, the left image data is encoded in the red channel and the right image data in the cyan channel. With the use of anaglyph glasses, each image is filtered to the respective eye, creating the perception of depth.

Although anaglyph 3D is successful in achieving stereoscopy and is the least expensive technique for mass 3D visualization, it manifests several problems like a faulty reproduction of color and crosstalk. The color rivalry between the eyes also increases the visual discomfort.

Several methods, such as image alignment, color component blurring, and depth map adjustment have proved to reduce crosstalk and improve image quality [28] [29].

2.2.1.2 Polarized 3D system

In Polarized 3D systems, the two overlying images are projected by a two-projector setup in which each projector produces one of the left or right images with orthogonal polarization to the other. The image is filtered by glasses with similarly polarized filters, allowing each eye to see a different image creating a perception of depth.

The glasses can be linearly polarized when the images are projected onto the screen via orthogonally polarized filters, or circularly polarized when the images are projected on the screen with opposite handedness (i.e., right-handed and left-handed).

Circularly polarized glasses are resistant to rotation and head motion and allow the viewer to tilt its head without increasing crosstalk. On linearly polarized glasses the orientation of the glasses departs from the polarization of the image, increasing crosstalk.

Polarized systems are capable of full-color representation, greatly reducing the visual discomfort. However, some crosstalk between the images remains. Furthermore, polarization filters block a significant amount of light and projection systems require a maintained alignment of the two polarized images, as any misalignment can cause eyestrain. It is also required the use of a special screen to preserve the state of polarization of the light.

There are cost effective solutions which allow the use of a single projector, with no alignment requirements, but these solutions show an increase of light loss.

In the sphere of flat displays, stereoscopy is achieved using alternate lines of pixels in reverse, half for one eye, and the other half for the other. The alternate rows are orthogonally polarized by the use of *micropolarizers* attached to the display. This results in the reduction of the available resolution in half [28] [29].

2.2.1.3 Active shutter 3D system

Active shutter 3D systems use liquid crystal (i.e., active) shutter glasses that alternatively block the image intended for one eye and display the other. This process is repeatedly rapidly enough, usually at a frequency of 120 Hz, to prevent interfering with the perceived fusion of the two images. Active shutter systems solve the image alignment and light output problems of Polarized 3D systems, but require synchronization between the glasses and the video signal, a larger video bandwidth and are more expensive than passive polarized glasses [28] [29].

2.2.1.4 Head-mounted display

In a similar way to stereoscopic systems, HMD display a different image to each eye with the use of a separate display and lenses to relax eye focus, positioned in front of each eye.

The two images can be displayed by different methods: separate video signals to each eye, frame sequential multiplexing - in a similar way to active shutters, the left, and right images are alternated in successive frames - and top/bottom multiplexing - half of the image is allocated to the right eye and the other half to the left.

HMD can support several additional features, such as eye tracking, for the measurement of the point of focus, and head tracking, which allows the user to control the image with the movement of the head. These features grant an enhanced and more immersive experience and prove advantageous for a range of applications from VR training and simulation to applications in engineering, science and medicine through the exploration of 360^{o} video and virtual reality environments.

One of the main problems of HMD is the latency between the movement of the head and the displayed content. There are also issues of visual discomfort due to the differences between the left and right images. Accommodation-convergence mismatch also occurs [29] [30] [31].

2.2.2 Autostereoscopic Methods

2.2.2.1 Two-view Displays

Parallax barrier displays

In Parallax barrier displays; there is an opaque layer of material (e.g., thin sheet of aluminum) with a series of narrow precision slits, regularly spaced, placed in front of the display. The slits are meticulous aligned with the pixel columns and work as microscopic projectors, allowing each eye to see a different set of pixels. This method grants the display the ability to show stereoscopic images without the requirement of any eyewear, thus creating *Autostereoscopy*.

Nonetheless, the perception of depth varies with the user's position, and it is mandatory to the user to be confined to a *viewing zone*, a well-defined spot at a certain distance from the screen where consistent 3D perception is experienced, otherwise, the viewer can perceive a *pseudoscopic image* (i.e., a type of crosstalk), which occurs when the displayed views are reversed. Parallax barrier displays also present reduced light output due to the opacity of the parallax barrier and show a severe cut on the horizontal resolution as the horizontal pixel count intended for each eye is reduced in half [28]. It is possible to eliminate the static *viewing zone* by using dynamic parallax barriers, but it results in a system with less resolution and poorer image quality.

Parallax barriers can also be used with projectors. Using multiple 2D projectors along with two parallax barriers, the first barrier controls the output of the projectors and the second works as a regular parallax barrier [29].

Lenticular displays

Lenticular displays typically have on top of a pixel addressable screen, a sheet of thin cylindrical *microlenses* precisely aligned with the pixel columns and designed to magnify and direct the light coming from the underlying panel, allowing a user at viewing distance to perceive depth. Lenticular displays do not block light, but they still require the user to be bound to the *viewing zone* and manifest the same problems of crosstalk and the reduction on the horizontal resolution of the parallax barrier displays. They are also more difficult to manufacture, hence more expensive [28]. Residual lens effects - a pattern of dark and bright bands observed at different angles as the viewer moves - also occurs at oblique angles [29].

Figure 2.2 illustrates the distinction between the two displaying methods described above.

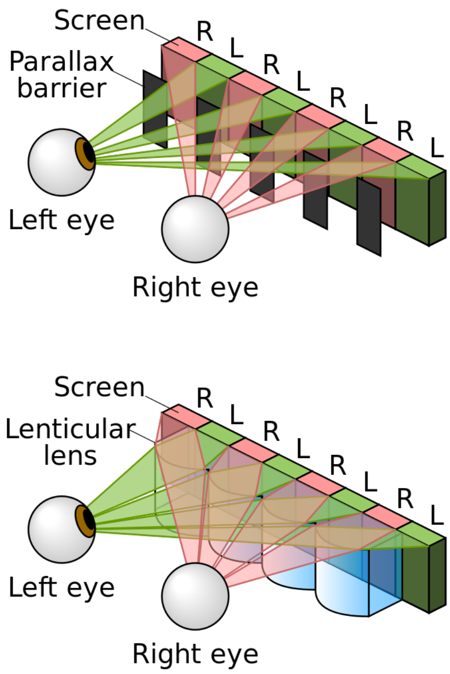

Figure 2.2: Autostereoscopic displays.
(Source: wikipedia.org, 2015 [32])

2.2.2.2 Multiview Displays

Multiview displays take in account motion parallax. The displayed view changes according to the viewer motion, providing an important cue for accurate depth perception.

Active Multiview Displays

Active multiview displays track the viewer's location, providing an appropriate view-pair from all the several viewpoints available. Active displays come close at displaying real 3D objects, taking advantage of stereoscopy and motion parallax, however, as the system requires the viewer's head to be tracked, it can only support single-viewer experiences [28].

Passive Multiview Displays

Passive multiview displays, also called, *Automultiscopic* displays, use spatial multiplexing to direct each view to a distinct viewing zone. All views are displayed at the same time. Thus, no head tracking is necessary, multiple viewers can experience 3D in their respective viewing zone and viewer motion between zones induces motion parallax. This effect is achieved using parallax barrier or lenticular displays, as such it is still mandatory for the viewer to remain at a specified distance from the screen.

Multiview displays manifest serious imbalances between vertical and horizontal resolution, as they provide horizontal parallax only [28].

2.2.2.3 Light field displays

A common type of Light field displays is the *Super Multiview display* (SMV). SMV is a type of multiview display capable of inducing an apparent continuous parallax by displaying a sufficiently large number of discrete images. When the angle pitch between two beams of light is between 0.2^o - 0.4^o, more than two rays pass through the same point in space and enter the viewer's eye simultaneously, providing more than two views for each eye. It has been proved possible to resolve the issues of visual fatigue induced by accommodation-convergence mismatch, by presenting 64 viewing zones.

Eye tracking can soften the load of displaying more than two views to each eye. If it is not performed, a large amount of information must be displayed.

SMV provides multiview and *multiviewer* capabilities, but also presents complex hardware and software requirements [29].

G. Wetzstein et al. [33] describe a group of compressive light field displays that offer significantly increased resolution and brightness when compared to conventional 3D displaying methods, granting superior image quality.

Also, In a recent article, P. Kovács et al. [34] try to measure the perceived resolution of light field displays, as such screens do not have discrete pixels. P. Kovács et al. show that horizontal resolution is slightly lower when perceived from the edge of the field of view. Additionally, it has also been shown that the viewer's head movement proved to improve the perception of fine details in the image, reinforcing the role of motion parallax on an accurate 3D image perception.

2.2.2.4 Holographic displays

Holographic displays represent the ideal technique to display 3D content, as it mimics the natural light emitted by physical objects, allowing extremely accurate 3D representation of the original scenes. Holographic displays reconstruct the wavefront of light, by spatially-varying intensity information along with phase information. Difficulties arise building this kind of displays due to the ultra high resolution required by the spatial light modulator, the device responsible for displaying the image.
When proper displaying conditions are met, accommodation-converge mismatch does not occur and smooth motion parallax is obtained [28] [35].

The transmission and storage of holographic content also prove to be a considerable challenge due to the large bandwidth required to transmit the huge amounts of holographic data. The bandwidth requirements approach the highest bandwidth provided by the available networks, up to 100 Gb/s and can even exceed future's network's capability, reaching up to 1 Tb/s. Solutions to achieve feasible data transmission include the compression of hologram data and the transmission of 3D object data instead of computed holograms, but they still manifest challenges on hardware and software requirements [36].

2.2.2.5 Volumetric displays

Several approaches such as rotation planes, static, holographic and multiview have been employed to create volumetric displays.

Volumetric displays appear to emit light from an entire volume of space, accurately reproducing the wavefronts curvature of light coming from an object.

Each point in the volume has its intensity modulated by an electrical signal and has an appearance independent of viewer location, avoiding the aliasing artifacts of multiview systems, as the view zone is not discretized. Still, volumetric displays cannot maintain occlusion cues for all viewers, limiting the array of images where volumetric displays excel on 3D representation [28].

The processing and transmission of information in volumetric displays also manifest huge bandwidth requirements, as shown by I. Osmanis et al. [37]. The required bandwidth can range from 6 Gb/s, with 16 b/sample with a refresh rate of 25 Hz to 18 Gb/s, with 24 b/sample at 50 Hz, for a resolution of 1024x768.

2.3 3D Coding Formats

The main focus of this section is to provide an overview on the state of the art of the compression methods related to 3D multiview video. This section begins by presenting a brief overview of the H.264/AVC standard. Performance results of the several methods are presented in order to identify the best fitting standard for the system under study.

2.3.1 H.264/AVC

H.264/AVC still represents the most common format for the compression of video content. H.264/AVC follows a block-based motion-compensation approach for the coding of video content. Some of the most important features include the use of small block-size exact-transform, adaptive *in-loop deblocking* filter, enhanced motion prediction and entropy coding methods.

One of the key features of H.264/AVC is the use of a *Network Abstraction Layer* (NAL) and a *Video Coding Layer* (VCL). The VCL is responsible for representing the video data, as described above. While the NAL is specified to format the data and provide header information for transport or storage. [38]

2.3.1.1 H.264/MVC

MVC is the multiview extension of the H.264/AVC standard, onward referred simply as AVC. Being an extension, MVC maintains compatibility with the AVC standard, allowing single views to be decoded by ordinary AVC decoders.

MVC stands out when coding multiview video, by performing disparity-compensated inter-view prediction using a block-based disparity shift between the reference view and the different views of the same scene at the same time instance. The MVC approach consists simply in providing high-level syntax to appropriate signaling view identifiers and their references and by defining a process for inter-view prediction [39].

MVC only proves to reduce the stream bitrate when using several video sequences or when employing buffering and smoothing, having no apparent advantages when coding ordinary 3D (i.e., two-view or stereoscopic) video content, thus proving appropriate for multiview 3D video coding [40].

2.3.1.2 MVC+D

MVC+D introduces depth support for multiview coding to enable depth-image-based rendering for all the viewpoints while maintaining compatibility with stereoscopic video streams. A second independent stream is used to represent both depth information and high-level signaling syntax [39].

2.3.1.3 3D-AVC

3D-AVC supports depth coding in a similar way to MVC+D and also adds support for new block-level coding tools for textures. Additional coding tools include inter-view motion prediction.

Taking advantage of the high levels of correlation of motion information between views, inter-view motion prediction grants eminent gains in coding efficiency and is achievable by calculating disparities using *neighboring block-based vector derivation* (NBDV), which derives disparity motion vectors of temporal and spatial neighboring blocks directly without the need for additional information. *View synthesis prediction* (VSP) is present and uses a block-based scheme for prediction using depth information to warp texture data from a reference view to the current view. Finally, support for illumination compensation is also present to improve prediction performance by compensating for color and lighting flaws during the capture process due to uncalibrated cameras, improving coding efficiency [39].

In [41] is described a cross-lab subjective evaluation test that shows that although 3D-AVC offers in average a 14% bit rate reduction when compared to MVC+D, using multiview video sequences, the confidence intervals, where the bitrate difference varies from -30% to +8% suggest that the MVC+D can sometimes offer better performance than the 3D-AVC. The rate-distortion curves of the two standards also suggest that generally, the difference between the two codecs is not significant.

2.3.2 H.265/HEVC

HEVC, the successor to AVC, was jointly developed by the ISO/IEC Moving Pictures Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG).

HEVC replaces the use of macroblocks for Coding Tree Units (CTU) with dimensions of 16x16 to 64x64 pixels. The use of CTU of a larger sizes proved to increase coding efficiency, mostly in very high resolutions. The CTU can be grouped in slices for better synchronization and tiles, for parallel processing. HEVC supports 33 directional orientations for intra-prediction and two modes for motion prediction, *Advanced Motion Vector Prediction* (AMVP) and Merge.

There are several profiles and levels defined to offer different performances to meet the applications requirements.

HEVC provides, for the same image quality, bitrate savings of approximately 50%; relative to the AVC; or can offer greater image quality for the same bitrate [42].

2.3.2.1 MV-HEVC

Similarly to the MVC standard of AVC, the multiview extension of HEVC, offers interview prediction and compatibility with single-view coding of HEVC (HEVC version 1), exploiting the advantages of the HEVC standard and allowing to achieve superior compression rates.

MV-HEVC only contains high-level syntax changes to enable reuse of HEVC version 1. As suggested in *"Motion Hooks for the Multiview Extension of HEVC"*, by Y.Chen et al. [43], any block-level operation necessary for MV-HEVC is enabled by motion hooks, i.e., motion prediction techniques fundamental for MV-HEVC, but that do not interfere with the first version of HEVC. Some of the proposed techniques, namely, *stable motion prediction* for interview references and *type oriented motion prediction* (TOMP), became part of the standard. The use of these techniques showed an average reduction of 4% in the bitrate on simulation using multiview video sequences [43].

Depth support for MV-HEVC is achieved with auxiliary picture syntax, enabling applications to decode depth maps optionally [39].

2.3.2.2 3D-HEVC

3D-HEVC also supports coding tools for textures as well as all the features present in the 3D-AVC, with some design changes.

Additional features of 3D-HEVC include *advanced residual prediction* (ARP), where the motion vector is aligned for both the reference block and current block, significantly reducing the resulting difference. 3D-HEVC also includes additional tools for depth coding, including *depth motion prediction* and *partition-based depth Intra coding*, applying two different types of partitioning patterns to better fit the particular characteristics of depth. Finally, the *segment-wise DC coding* (SDC) mode allows the transform and quantization process to be skipped, so depth prediction residuals are directly coded. SDC can be applied to Intra and inter prediction [39].

3D-HEVC is capable of encoding 3D videos with or without depth maps associated to texture frames.

A quality analysis performed by Balota, G. [44] et al., which compared the performance of MVC and 3D-HEVC using multiview video sequences, shows that 3D-HEVC can achieve a reduction of more than 50% on the bitrate for the same quality results.

2.4 Free-viewpoint Television and 360º Video

2.4.1 Free-viewpoint Television

Free-viewpoint television (FTV) refers to any system capable of displaying a 3D video and allowing the user to freely switch between viewpoints.

FTV provides a natural interface between the user and the environment and allows the creation of many types of content and a broad range of application from entertainment to education, medicine, and surveillance [45]. Examples of FTV systems are active multiview displays and SMVs.

FTV standardization is underway and began with the creation of the MVC extension of the AVC standard. Currently on its third phase, FTV standardization aims to offer a very wide viewing space, horizontal and vertical parallax, smooth transition between adjacent views and motion parallax, reduced eye fatigue and free navigation of 3D scenes [46].

2.4.2 360º Video

Concurrently, new forms of interactive and immersive video displaying have emerged. 360º video consists on high-quality and seamless 360-degree panoramic video. The panoramic video can be obtained by the use of special lenses, such as fisheye lenses, and or by multi-shot methods. Theses methods are based on technologies for stitching a set of images with overlapping regions, captured by a set of multiple cameras [47].

There are several solutions for capturing, producing and playing 360º videos in HD and UHD. Some applications find the use of HMD devices fit for displaying immersive 3D 360º videos. There are also solutions available for 360º video streaming. 360º video differs from FTV in the form the video is perceived from a first person point-of-view, centered on the viewer [48] [49] [50] [51].

2.5 Adaptive Streaming Techniques - MPEG-DASH

Video streaming over the Internet is an ambitious task due to the nature of IP networks. The streaming methods have to take in account the variable availability of network bandwidth, the fact that prediction techniques employed in video compression create dependencies on several data packets - making the client vulnerable to missing data - and the heterogeneity of client devices [52].

This section highlights the MPEG-DASH streaming technique but also contains a brief overview of the scalable extensions of AVC and HEVC.

2.5.1 SVC

SVC is an extension of AVC that supports Scalable Video Coding (SVC) by providing a base layer with a basic version of the video, along with enhancement layers. Enhanced video layers can drop packets to match bandwidth requirements, offering video of lower quality in a situation of network congestion. The reduction of quality can be obtained by reducing temporal resolution, spatial resolution or SNR video quality.

Although SVC appears to be fitting for video streaming, it presents a relatively high overhead when more than two enhancement layers are used [40].

2.5.2 SHVC

The Scalable High Efficiency Video Coding is the scalable extension of the HEVC, similarly to the SVC extension of the AVC.

SHVC provides spatial and quality scalability by using a multi-loop decoding structure, where all layers are decoded. Therefore, SHVC consists of a group of HEVC encoders, having one encoder for the base layer and the remainder encoders for the enhancement layers.

Although inter-layer prediction improves the coding efficiency by enhancing the rate-distortion performance by 15% to 30%, SHVC introduces an additional coding complexity - in average, decoding two-layers introduces 43% to 80% of additional complexity in comparison to a single-layer HEVC decoder [53].

2.5.3 MPEG-DASH

Dynamic Adaptive Streaming over HTTP or MPEG-DASH, takes advantage of the extensive support for HTTP on the Internet infrastructure, allowing HTTP streaming to be highly cost-effective.

With HTTP streaming, the client manages the streaming without the need to maintain a session state on the server and decides which and when media data is downloaded, allowing *Content Delivery Networks* (CDN) to provide a large number of clients, without additional costs on the servers.

In general, in MPEG-DASH, both the media and signaling metadata are delivered by HTTP. The content stored on an HTTP server consists on a *Media Presentation Description* (MPD) and on segments. The MPD is an XML document which describes a manifest of the available content (i.e., the metadata).

The DASH client obtains and parses the MPD, learning about the program characteristics and available representations (i.e., encoding alternatives), then selects the appropriate representation for streaming and finally starts fetching the subsequent segments for a specific period. Buffering is required in order to account for network throughput variations.

The client monitors the network bandwidth fluctuations and decides how to adapt the available bandwidth by fetching segments of different alternatives.

At the end of every period, the client obtains the MPD for the next period, the whole process is repeated.

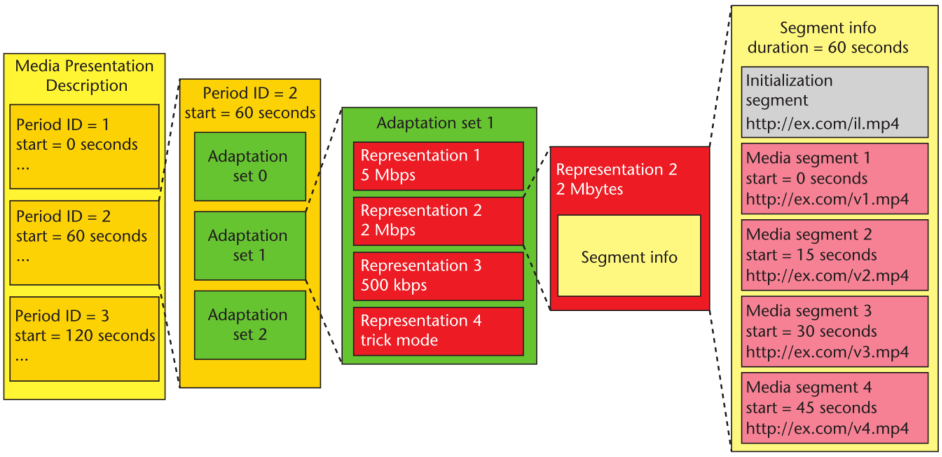

Figure 2.3: MPD data model
(Source: Industry and Standards The MPEG-DASH Standard for Multimedia Streaming Over the Internet, 2011 [54])

The MPD data model is divided into periods, with a starting time and duration. A period contains a temporal chapter of the media component. Each period incorporates an adaptation set, containing information of all the available representations (i.e., the available alternatives of the same multimedia content). Representations may represent variations of the same media content on bitrate, resolution, number of channels or any other characteristic.

Each representation consists of one or multiple segments. Segments represent the media stream and each segment contains a unique URL to its respective location on the server, and index, a start time, and duration. The first segment is the initialization segment and does not contain any media data. Each segment contains one access point in order to allow the client to start decoding data at any point of the stream.

Segments are delivered by a sequence of HTTP request-response transactions, over either UDP or TCP transport. For live streaming, the segments contain additional information, including segment availability. The interval between two requests is close to the duration of the previous segment. It is suggested an initial buffering of about 2 segments [54] [55]. Figure 3.2 shows the MPD data model described.

MPEG-DASH supports digital rights management (DRM) and a vast set of additional features, including the ability to select and switch streams (e.g.: the client can select a different view of a multiview video sequence or an audio stream of a different language). MPEG-DASH offers support for SVC and MVC and multiple base URLs. A thorough description of MPEG-DASH and all the available additional features can be found in [54].

The use of SVC has proven to improve the storage and caching efficiency for DASH streaming [55].

2.5.4 Quality of Service and Quality of Experience

In order to ensure the best quality of service and experience possible, an MPEG-DASH system must deliver high-quality video while providing low latency. In the system under study, where several views are available and the user can change between views at any moment, it must be guaranteed latency remains minimal, otherwise, it can severally degenerate the user's experience. To assure low latency is achieved, the video quality may be compromised due to bandwidth fluctuations.

Latency in MPEG-DASH is affected by the segmentation delay; the asynchronous fetch of media segments; the time to download the segments; and the buffering at the client side [56]. A study by Bouzakaria, Nassima et al. [56], using the GPAC framework, shows that live streaming of HD video samples with an encoding overhead of 13%, when applying the gradual decoding refresh, chunked-transfer encoding, and ISOBMF packaging techniques in order to minimize the overall latency, can achieve an end-to-end latency in local networks as low as 240 ms, thus proving that very low latency streaming is possible in local networks.

Additionally, it is possible to provide seamless streaming when bandwidth fluctuations occur. Lee, Sunghee et al. [57] propose the Video Quality Control for QoE (VQCQ) scheme that provides smoother changes of video quality by calculating the optimal video quality using the buffer status and the download speed of the next segment, outperforming the concurrent RAHS and ASAS schemes.

On *"A DASH-Based 3D Multi-view Video Rate Control System"* [58], it is proposed a DASH-based multiview streaming system, similar to the system under study, using a 3D-HEVC extension test model. The system is tested under different network conditions and is also subjected to a subjective test. This system allows the transmission of different 3D video sequences in higher quality, compared to a non-dynamic adaptive system. Using test video sequences with 2 and 3 views, the authors, Su, Tianyu et al. conclude that 2-view videos provide better video quality than the 3-view videos, at any given bitrate. Also, and as expected, as the bitrate increases, the perceived quality also increases. Finally, the authors show that the minimum bitrate required for good video quality should be not less than 1.2 Mb/s, with 2 views, and 3 Mb/s with 3 views.

2.6 Machine Learning

Machine Learning is a subfield of artificial intelligence which explores the construction of algorithms that can automatically learn from data and make predictions based on past observations [59].

This section begins with a classification of the types of machine learning algorithms according to the input data and model preparation process, as found in [60]. It follows with an exploration of the algorithm selection process and concludes with the subsequent selection and brief description of the most suitable algorithms.

2.6.1 Learning Style

Supervised learning

Supervised learning algorithms build a model through a training process. The algorithm uses labeled input data, called training data, to make predictions and consequent corrections until a certain level of accuracy is achieved. This type of algorithms are used in classification and regression problems [60].

Unsupervised learning

Unsupervised learning algorithms do not use labeled data and do not produce a known response. The models are built by deducing structures from the input data. These types of algorithms are used in association rule learning and clustering problems [60].

Semi-supervised learning

Semi-supervised learning algorithms use a set of mixed labeled and unlabeled data in order to build the structures to organize the data and make predictions. These type of algorithms are used as extensions to other flexible methods and in classification and regression problems [60].

Reinforcement learning

In reinforcement learning algorithms, the input data is provided in a feedback scheme in order to provide stimulus to a model who must respond and react to a certain environment. These types of algorithms are used in control systems and robot control [60].

2.6.2 Supervised Learning Algorithms

Due to the nature of the addressed problem, the only learning style that should be taken into consideration for use in the prototype is supervised learning. As formerly described, supervised learning algorithms use a set of input data (i.e., training data) and known responses to the data in order to build a model capable of predicting responses with a certain level of accuracy, as shown in Figure 2.4.

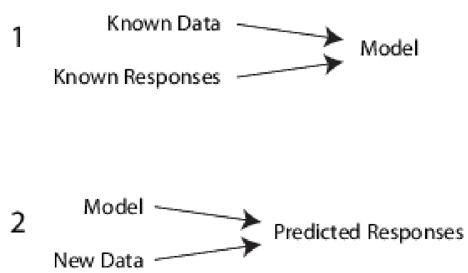

Figure 2.4: Supervised learning algorithms
(Source: mathworks.com, 2015 [61])

Supervised learning can be applied into classification problems when the responses are limited to a certain number of known values (e.g., 'true' or 'false') or regression problems when the responses are represented by real numbers.

After the identification of the type of problem and before the selection of the algorithm, it is necessary to prepare the data for the input.

The key characteristics that limit the choice of the algorithm are the speed of training, the memory usage, the predictive accuracy of the new data and the interoperability (i.e., how easily can the predictions be understood). Characteristics comparisons of the different algorithms are available at [61] and [62]. It is possible to conclude, with the available data, that the SVM algorithm can offer better performance - if few support vectors are used - by achieving high predictive accuracy, low memory requirements, fast prediction speed and adequate training speed.

In order to verify the fitness of the SVM, it is recommended to test the performance of different algorithms. The characteristics of neural networks and decision trees suggest these algorithms are good alternatives to the SVM if the ideal conditions for SVM prediction are not met.

2.6.2.1 Support Vector Machine

Support Vector Machines (SVMs) are used in linear classification problems when the data has exactly two classes. The data points in each class are classified in order find the largest margin between classes, (i.e., the best hyperplane). The best hyperplane separates all data points of one class from the other. The data points closest to the hyperplane represent the support vectors.

When non-separable data is present, SVM uses a *soft margin*, to separate the majority of the data points. It is possible to use Kernels for problems that do not have simple hyperplanes to obtain nonlinear classifiers, without increasing the complexity of the calculations [63].

2.6.2.2 Decision Trees

Decision Trees can be applied to classification and regression problems. The prediction of responses follows the decision in the tree from the root (top) node to a leaf node (response). In each step, the value of the predictor is evaluated in order to identify which branch to follow. When the leaf node is reached a response is obtained [64].

2.6.2.3 Neural Networks

Neural Networks can be applied to classification and regression problems in supervised learning, but also find applications in unsupervised learning and reinforcement learning.

The building blocks of neural networks are the single-input neurons as shown in Figure 2.5. The three processes that encompass the simple neuron are the weight function (wp), the net input function (n) and the transfer function (f), where p represents the input. p can be a scalar or a vector.

The scalar parameters b and w are adjustable, making it is possible to train the networks to perform as desired [65].

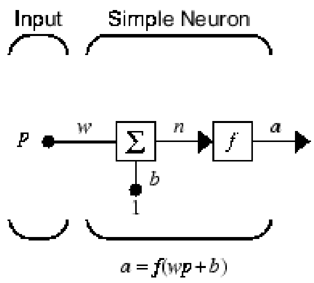

Figure 2.5: Neuron model
(Source: mathworks.com, 2015 [65])

With the combination of several neurons is possible to create a neural network used to obtain responses to different inputs. Neural networks can be interconnected on several different structures, from one-layer to multiple-layer networks [66].

2.7 Summary

The main obstacle to the acceptance of 3D is the visual discomfort introduced by unnatural viewing conditions and the requirement of glasses. Among stereoscopic methods, active shutter systems show the most promise in displaying 3D images in a comfortable way. HMDs also show great promise with additional features to provide an enhanced experience. Nonetheless, these systems still require the use of glasses or headgear and have stricter technical requirements. HMDs also manifests issues of visual discomfort.

Volumetric and holographic displays present huge technological limitations and cannot find a feasible application. SMV appear as the ideal solution for FTV, but the lack of availability and technological limitations makes impractical the use this kind of displays. Possible solutions for FTV point to the use of active multiview displays. A simple lenticular display and a web camera can be presented as a lower-cost solution.

Although samples coded in HEVC, can achieve bitrates saving of 50% and offer video of very high quality. AVC stills represents the most common video compression format. The multiview and 3D extensions can offer reductions in the bitrate, making possible the streaming of 3D content, with ease. It is expected HEVC will become the state of the art in video compression in the near-future. For 3D content, the compression gains of HEVC can be maximized using the multiview and 3D extensions.

FTV standardization is still in progress. The current lack guidelines play an important role in building efficient and reliable FTV systems. On the other hand, there are already multiple solutions for displaying 360º video. 360º shows huge promise in displaying new forms of interactive and immersive content. Together with HMDs, they can offer cost-effective solutions for viewing 3D and explore new forms of content.

Among adaptive streaming methods, MPEG-DASH is the state of the art for video streaming in IP networks. MPEG-DASH allows effortless delivery of segments and seamless streaming. The client manages the streaming, allowing CDNs to provide a large number of clients, without an additional load on the servers. MPEG-DASH can offer support for SVC and MVC and the use of SVC can improve the storage and caching efficacy of the streaming. However, the overhead and coding complexity introduced by the scalable extensions of AVC and HEVC may introduce an undesirable overload in the systems. An MPEG-DASH streaming system can offer cost-effective streaming solutions, with efficient use of network resources.

Chapter 3

Methodology

This chapter presents the dissertation objectives, the followed approach and a description of the test scenarios.

3.1 Objectives

The goal of this dissertation is to finish and perfect the prototype and to achieve a fully-functional FTV system, applicable to a different range of scenarios of VOD streaming. Figure 3.1 shows the high level concept of the prototype.

The prototype uses MPEG-DASH for streaming, an HTTP server contains the media segments and the MPDs of the multiple views.

The client encompasses the player and the client application. The two modules communicate via UDP channels. The player manages the streaming, displays the content and handles the switching operations requested by the client application.

The client application uses a web camera to track the user head-movement in real-time and detect the focus of attention. If significant changes are detected, the client signals the player to request a new view to the server. The client application receives or monitors metadata and performance parameters.

By monitoring the network conditions, the client adapts the content (i.e., video quality) to bandwidth availability.

A cost-effective solution that provides the best quality of experience possible and takes into account network and terminal constraints should be obtained.

Problem: The prototype does not support streaming, the video quality selection is performed manually, and the transition between views introduces visible artifacts and does not run smoothly. It is desired that the prototype can adapt the video quality to the bandwidth availability.

As such, the specific objectives of this dissertation are as follows:

1. Literature review of the concepts and the most significant topics regarding the addressed problem, from 3D viewing and coding technologies to MPEG-DASH;

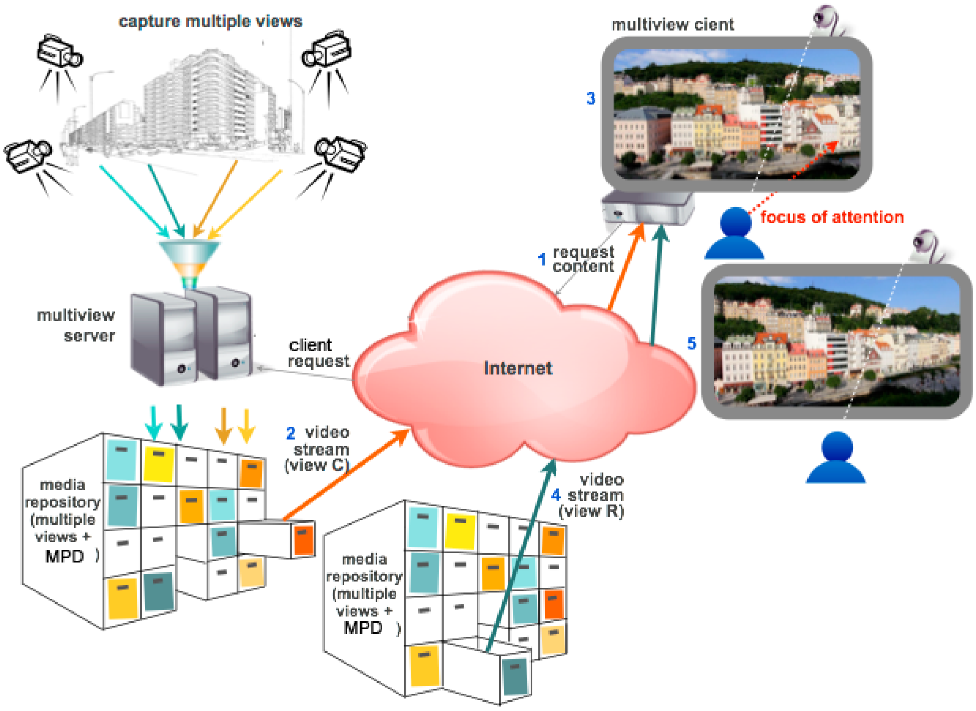

Figure 3.1: High level concept

2. Development, implementation and integration of a mechanism to allow a seamless transition between different views;

3. Implementation and integration of a mechanism for automatic adaptation of video quality (very low, low, medium, very high) to network conditions;

4. Improvement of system performance and QoE: Identification of latency sources and reduce latency.

3.2 Approach

The initial version of the prototype is in a semi-functional state and manifests several problems that must be resolved before addressing the dissertation objectives.

The project objectives are addressed in chronological order. Test scenarios are designed and a set of simulations, tests and analysis are performed to evaluate the current state of video streaming, the applicability of the prototype to real world environment, the quality of service and the quality of experience. Beyond these, the tests intend to find the ideal control parameters that allow the prototype to achieve optimal performance and compare the prototype and its features with common streaming services.

The test scenarios are further described in this chapter in section 3.3.2.

With the intention of providing an accurate and clear description of the developed work, the prototype problems are addressed in 4, as part of the prototype description. The dissertation objectives are grouped into the two different categories, *Quality of Service* and *Quality of Experience* and described in chapters 5 and 6, respectively.

3.2.1 Definition of Priorities

A different priority was assigned to each problem according to its role in the operation of the prototype. The priority values are qualitative, representing a level of urgency, and are defined as the follows:

- **Low**: Problem does no interface with the operation of the prototype nor with the objectives of the dissertation;
- **Medium**: Problem hinders the performance, but does not interface with the operation of the prototype, nor with the objectives of dissertation;
- **High**: Problem interferences with the operation of the prototype or with the objectives of the dissertation.

In conclusion, problems with *High* priority affect the proper operation or the QoS and consequently, the quality of experience, whilst problems with *Low* and *Medium* priorities only affect the QoE.

3.2.2 System Versioning

Due to the high number of changes made to the prototype during the course of this dissertation, the following version numbers were assigned to each state of the prototype to better describe the two stages of development:

- **Version 1**: State of prototype at the beginning of the dissertation;
- **Version 2**: State of the prototype at the end of the dissertation.

Henceforward, each state of the prototype will be referred to its version number.

3.2.3 QoS Parameters

Although QoS is the concern of CDNs, the prototype must guarantee that QoS is offered and should adapt the video quality to network availability in scenarios of limited network capacity, network fluctuations and low-throughput situations.

Two parameters are used at the application level to evaluate the prototype performance and QoS: The latency and the percentage of lost frames.

Four types of latency are identified:

- **Streaming latency**: Latency between the fetching of segments and the decoding and displaying of the segments. This latency is compensated by the utilization of a buffer, as defined on the MPEG-DASH specification [67]. Streaming latency should remain minimal and any throughput variation should be accounted for by the buffer [54]. The streaming latency is independent of any latency introduced by the switching or head tracking operation;
- **Switching latency**: Latency of the switching operation, from the time the operation is requested on the client application to the time the player displays a new representation. This is the total latency in manual mode;
- **Head tracking latency**: Latency of the head tracking operation, from the time a head movement occurs to the time a view switching operations is requested;
- **Automatic mode latency**: The value of the *switching latency* plus the *head tracking latency*.

The percentage of lost frames is computing using metadata obtained from the player. This process is further explored in chapter 4 and 5.

3.2.4 Work Plan

The following work plan was followed:

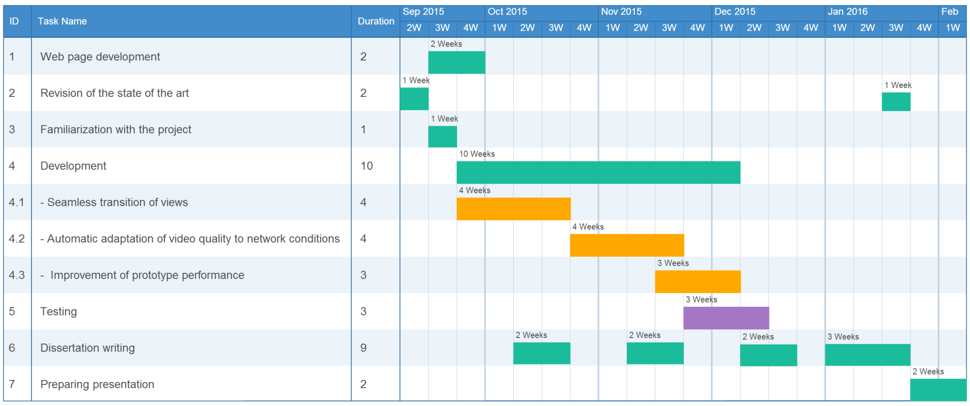

Figure 3.2: Work plan

1. **Website development**: Slot reserved for the website development.
2. **Revision of the State of the Art**: A revision of the state of the art was performed at the beginning and before the delivery of the dissertation to ensure that all the information is up-to-date and state of the art technologies were used whenever possible;
3. **Familiarization with the project**: Due to the complex nature of the project, a two-week slot was reserved for a proper adaptation to the workplace, the work tools and to acquire all the necessary information in order to better understand the project and its objectives. As the starting date of the dissertation was postponed, this slot was reduced to one week;

4. **Development**: The purpose of the development stage was to obtain a solution for each objective within the proposed time frame, the solution may vary depending on the problem and was achieved through the development and implementation of a new feature, through modifications of previous solutions or by the use of available tools. The changes and improvements made to the prototype are no listed and were performed wherever they interfered with an objective;

5. **Testing**: The testing stage consisted on designing and performing all the necessary tests in order to obtain a proper evaluation of all the developed features. Some of the tests were performed during the course of other tasks;

6. **Dissertation Writing**: Slot reserved to the writing of this document;

7. **Preparing Presentation**: Slot reserved for preparing the presentation of the dissertation.

The total duration of this dissertation was shorter than initially planned. Some of the time slots had to be adapted to the new due date.

3.2.5 Technologies, Tools; and Work Platforms

The prototype uses MPEG-DASH for streaming. For compatibility purposes, HEVC cannot be used and the media streams are coded in MVC. Due to the unavailability of lenticular displays, an ordinary display is used, along with a Web camera for head tracking. Each view is displayed side-by-side in a single stream.

The GPAC framework

The prototype uses the MPEG-DASH standard in order to deliver high-quality streaming from an ordinary HTTP server and was built on top of GPAC software.

> "GPAC is an open-source software dedicated to rich-media and broadcast technologies." [68]

GPAC provides a range of tools and options for manipulation, distribution and playing of media content. The packages and modules used in this project are addressed in chapter 4.

The HTTP Server

The server was set using the NPM package ecosystem from Node.js. Node.js was chosen because it allows to quickly and easily set up a reliable HTTP Server on any system.

> 'Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world." [69]

Work platforms and programming languages

The head tracking information is captured with a PlayStation Eye web camera and provided by the FaceAPI recognition software [70]. The GPAC framework is written in C++ and FaceAPI is only available on Windows platforms. As such, all the work was performed over the CPAG framework in the C/C++ programming languages and over the client application in C#. The IDE of choice was Visual Studio 2015 Community, running on Windows 10.

Documentation

All the documentation was created in LaTeX using ShareLaTeX. ShareLaTeX is an online, collaborative LaTeX editor [71].

3.3 Test Scenarios

The following test scenarios were designed to evaluate the performance of the prototype from local storage (for comparison purposes) and in real world environments.

3.3.1 Simulation of the Quality Adaptation Algorithm Behaviour

3.3.1.1 Simulation 1: Expected behaviour of the quality adaptation algorithm in real-world environments

This simulation analyses two different scenarios:

1. **Scenario A**: Expected behaviour of the quality adaptation algorithm in a common streaming service;
2. **Scenario B**: Expected behaviour of the of the quality adaptation algorithm in the prototype.

The quality parameters of the quality adaptation algorithm described in section 5.2.2 are adjusted using average real world network bandwidth values and analyzed on low-throughput situations. This simulation is theoretical and iterative. Using a spreadsheet, the quality parameters are adjusted to find the theoretical values that guarantee QoS for the widest range of network scenarios.

Objectives

1. Identification of average bandwidth requirements for video streaming of increasing quality;
2. Identification of network congestion upper limits insufficient to support video streaming of superior quality;
3. Prediction of the behaviour of the adaption algorithm on common streaming services in real-world environments;

4. Identification of upper and lower limits of network utilization that trigger the adaptation algorithm, in real world environments.

Expected results

1. Achievement of the best network utilization parameters to be used in the adaptation algorithm;

2. Achievement of the best bandwidth and congestion values to be used in the performance tests.

3.3.2 Performance Tests

3.3.2.1 Test 1: Performance evaluation of a presentation from local storage

This test evaluates the performance of the prototype and provides reference values for the following test scenarios.

The testbed shown in table 3.1 is used.

Table 3.1: Test device specification

Device	Surface Pro
OS	Windows 10 Pro x64
Processor	Intel i5-3317U 1.70 GHz
Memory	4 GiB

Objectives

1. Measurement of the prototype frame loss;

2. Measurement of the prototype initial buffering duration;

3. Measurement of the prototype head tracking latency;

4. Measurement of the prototype switching latency.

Expected results

1. Achievement of the prototype head tracking latency values to be used as reference in *Test Scenario 2*;

2. Achievement of an overview of the prototype performance in presentation from local storage.

Table 3.2: Test devices specifications

Role	Server	Client
Device	Desktop	Surface Pro
OS	Windows 10 Pro x64	Windows 10 Pro x64
Processor	Intel i7-2600 3.40 GHz	Intel i5-3317U 1.70 GHz
Memory	8 GiB	4 GiB

3.3.2.2 Test Scenario 2: Performance evaluation when streaming from an HTTP server on a Client-Server Architecture using real-world specifications

The testbed shown in tables 3.2 and 3.3 is used.

As the range is not taken into account, a 5 GHz frequency is used because it suffers less interference than the 2,4 GHz frequency [72]. The network throughput was measured with *iPerf* [73] using a TCP connections with the duration of 30 s. The RTT and loss were measured with the ping command using 60 packets.

Network congestion was achieved using *NetLimiter 4* [74] using the configuration presented in Table 3.4.

Objectives

1. Measurement of the prototype frame loss in simulated real-world environments;
2. Identification of the loss threshold that hinders the prototype performance in real-world environments;
3. Evaluation of the prototype behaviour in limited network capacity or low-throughput situations;
4. Measurement of the prototype initial buffering duration in simulated real-world environments;
5. Measurement of the prototype switching latency in simulated real world environments.

Expected results

1. Achievement of the best loss threshold value, to be used in the adaption algorithm;
2. Achievement of an overview of the prototype performance when streaming from an HTTP-Server.

3.3.3 Quality Tests

3.3.3.1 Test 1: QoE evaluation of a presentation

This test analyses three different scenarios:

Table 3.3: Network specification

Communication type	5 GHz wireless network
Device	Linksys WRT610N v2
Throughput	24.4 Mb/s
RTT (Min.)	2 ms
RTT (Max.)	3 ms
RTT (Avg.)	2 ms
Loss	0 %

1. **Scenario A**: Presentation from local storage;
2. **Scenario B**: Presentation from an HTTP server on a client-server architecture in low-loss situations;
3. **Scenario C**: Presentation from an HTTP server on a client-server architecture in high-loss situations.

In this test a qualitative evaluation of the prototype QoE in manual and automatic mode is performed. This is a subjective video quality evaluation loosely based on the recommendation of the *"ITU-T P. 913: Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment"* that uses an ACR scale [75]. A single number in the range 1 to 5 is assigned to the different parameters, where 1 is lowest perceived quality, and 5 is the highest perceived quality measurement. This test is merely subjective and was performed by the author of this document as a suggestive description of the general experience offered by the prototype. This evaluation was performed in pair with the performance tests. The same testbeds described in *Test Scenario 1* and *Test Scenario 2* are used. The quality adaption algorithm controls the video quality. The view switching is random, according to user's will.

Evaluated parameters

1. Seamless streaming;
2. Perceived video quality;
3. User friendly input:
   (a) In manual mode refers to intuitive manual input;
   (b) In automatic mode refers to uninterrupted, intuitive and flawless head tracking detection.
4. Seamless view switching;
5. Seamless quality switching.

Table 3.4: *NetLimiter* configuration

Type	"Limit"
Direction	"In"
Limit type	"Connection"
Zone	"Local network"

Objectives

1. Subjective evaluation of the QoE offered by the prototype in a presentation from the local storage;

2. Subjective evaluation of the QoE offered by the prototype when streaming in real world environments.

Expected results

1. Overview of the general behaviour offered QoE by the prototype;

2. Achievement of comparative evaluation of the QoE offered by the prototype for in real world environments.

Chapter 4

Prototype Overview

This chapter describes the prototype and stresses the differences between versions 1 and 2 of the prototype. It begins with a functional specification, followed by a description of the architecture of version 1, highlighting the problems identified on the different modules. It follows a brief outline of the changes and improvements made on version 2.

4.1 Functional Specification

4.1.1 Actors

Three actors are identified:

- **Content Provider**: The *Content Provider* (CP) is responsible for managing and storing the media content on a CDN;
- **CDN Operators**: The *CDN Operators* (CDN) are in charge of delivering the content of the *Content Provider* to the *End Users*. CDN networks are responsible for offering a safe and successful delivery of content and providing a good QoS at transport level;
- **End User**: The *End User* (EU) is the content consumer. The *End User* visualizes and interacts with the media content.

4.1.2 Use Case Model

Figure 4.1 presents an overview of the prototype use case model, followed by a brief description of the prototype uses cases. The final version of the prototype should be able to support all the uses cases described.

Insert and modify content

Actor: CP

The CP can insert, replace or delete media streams in the server. The CP can generate the MPD and media segments for any MP4 file.

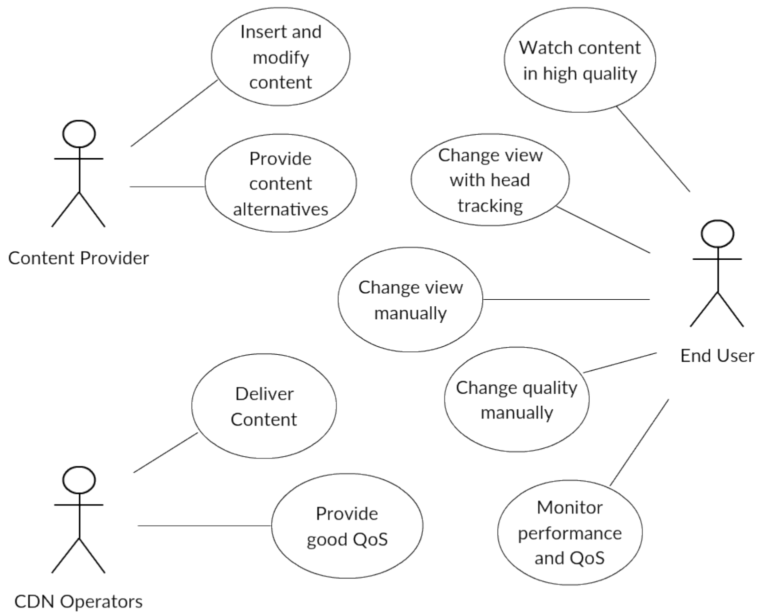

Figure 4.1: Use case diagram

Provide content alternatives

Actor: CP

The CP can provide several encoding alternatives, insert new view sets and new quality modes.

Deliver content

Actor: CDN

The CDN operator can effortlessly deliver content of every encoding alternative.

Provide good QoS

Actor: CDN

The CDN operator can provide good QoS for every encoding alternative.

Watch content in high quality

Actor: EU

The EU can watch content in high quality and enjoy a good QoE.

Change view with head tracking

Actor: CP

The EU can switch views with head movement.

Change view manually

Actor: CP

The EU has control over the QoE at application level and can switch views manually.

Change quality manually

Actor: CP

The EU has control over the QoE at the application level. The EU can change the video quality manually and influence the QoS.

Monitor performance and QoS

Actor: CP

The EU can monitor the client performance and the QoS.

4.1.3 Modular Architecture and Module Interaction

Figure 4.2 shows the modular architecture of the prototype and the interaction of between the different components.

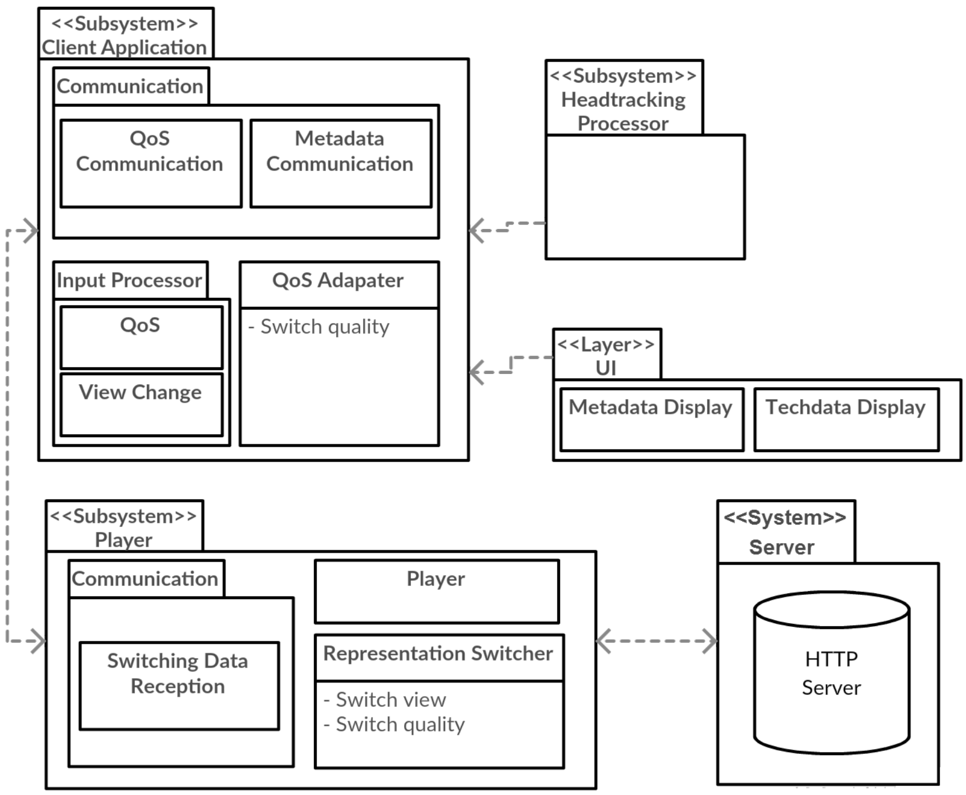

Figure 4.2: Modular architecture

The streaming and the prototype behaviour are controlled by the client machine on the EU side.

The client machine incorporates two subsystems: the *Player* and the *Client Application*. Communications channels between the systems ensure the exchange of switching data and metadata.

The *Player* manages the streaming and performs the switching operations requested by the client. The player also obtains and formats the metadata to send to the client application.

The *Client Application* displays *Metadata* and *Techdata* (i.e., performance and networking information). The client application uses a head tracking subsystem to support automatic view selection. An input processor grants support for manual view and quality selection. The client application monitors and controls the QoS at the application level, by automatically adapting the video quality to bandwidth availability.

On the CP side, a simple HTTP-server hosts the media streams and the MPD on a single-server distribution. On a CDN distribution, the server can host an MPD containing a description and links to the media streams hosted in other servers of the CDN.

The interaction between the different components is at the application level and is transparent to the CDN operators. The CDN operators should be able to provide good QoS.

The prototype should be able to work and provide seamless streaming and good QoE in different machines and network conditions.

4.2 Version 1

4.2.1 Architecture

Figure 4.3 shows the architecture of version 1.

4.2.1.1 The Media Streams

The media content consists of a scene shot in 3D with an array of 2D cameras placed alongside in arch, from 3 different angles, with some constraints. The maximum spacing between cameras should be of 65 mm or 1/30 of the distance to the object being filmed. Three different media stream (i.e, a different views) of the same scene were captured: *Left*, *Central*, and *Right*.

Each view is coded in four different bitrates, originating samples of different qualities: *Very Low*, *Low*, *Medium* and *Very High*, comprising a total of twelve different media streams. All the samples were coded using MVC in the MP4 format using *FFmpeg* [76].

The media stream specification is presented on table 4.1. The same bitrate is used for the three views.

Table 4.1: Media stream specification

Quality	Resolution (width x height)	Video codec	Frame rate (f/s)	Bitrate (kb/s)
Very Low	1920x1080	mp4a.40.2	25	1112
Low	1920x1080	mp4a.40.2	25	1613
Medium	1920x1080	mp4a.40.2	25	2680
Very High	1920x1080	mp4a.40.2	25	3739

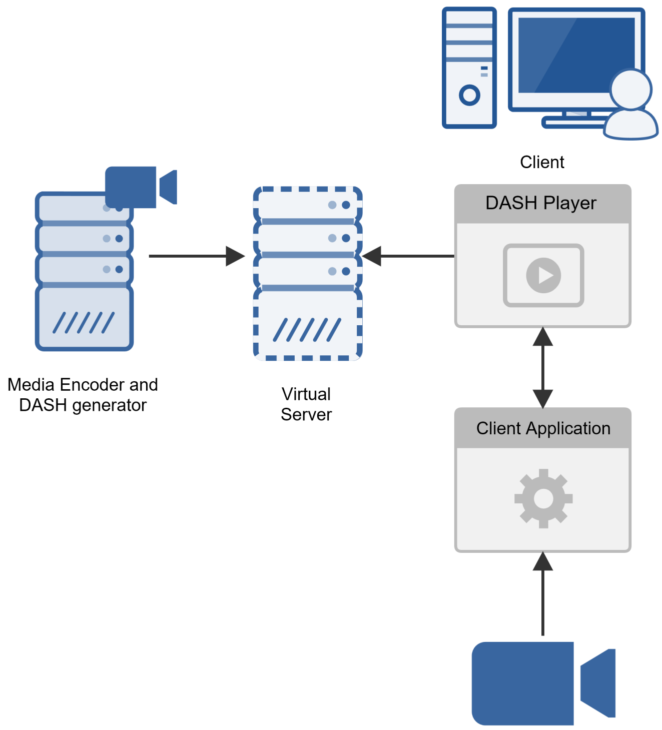

Figure 4.3: Version 1 architecture

4.2.1.2 The Media Encoder and DASH Generator

With VOD applications in mind, the prototype uses MP4Box for the generation of the media streams. MP4Box is a multimedia packager of the GPAC framework for the preparation of multimedia files for playback that supports the MPEG-DASH specification [77]. The content is first generated with MP4Box and then hosted on an HTTP server, for VOD streaming.

4.2.1.3 The HTTP Server

The server consists on the MPD and the media segments to be stored on a CDN distribution or on a single-server distribution. In the prototype, a single server distribution is used.

Problem: Version 1 does not support streaming. It manifests several problems when the media streams are hosted on an HTTP server, as such, there's only support for presentations from the local storage. The identified streaming problems are:

1. Automatic selection of the last available stream prevents the proper interaction with the media streams and generates odd behavior and poor performance.
2. Severe content loss and inability to decode some of the streams.
3. Significant switching latency.

The origin of these problems is on content generation operation and the player configuration.

4.2.1.4 The Dash Player

In HTTP streaming, the client manages the streaming without the need to maintain a session state on the server. The client obtains and parses the MPD and selects the appropriate representation. The client is responsible for controlling the automatic adaption of content to the bandwidth availability by monitoring the network bandwidth fluctuations and appropriately adapting the content by fetching segments of different representations.

Osmo/mp4Client is the multimedia player of the GPAC. It supports several media types, decoders, streaming techniques, including MPEG-DASH; and advanced features [78] [79]. Osmo includes a GUI, whilst MP4client uses a command-line interface.

In the prototype, MP4Client is responsible for the streaming. MP4Client was modified to and support and manage all the switching operations requested by the client. A configuration file allows to configure the MP4Client streaming options [80].

4.2.1.5 The Client Application

The client application establishes a connection with the dash player and provides an interface for switching between different views manually or automatically (i.e., by head tracking). It also supports a feature for manual selection of video quality. The head tracking information is displayed in real time on the application and control messages exchanged between the player and the application is also displayed, as depicted in Figure 4.4.

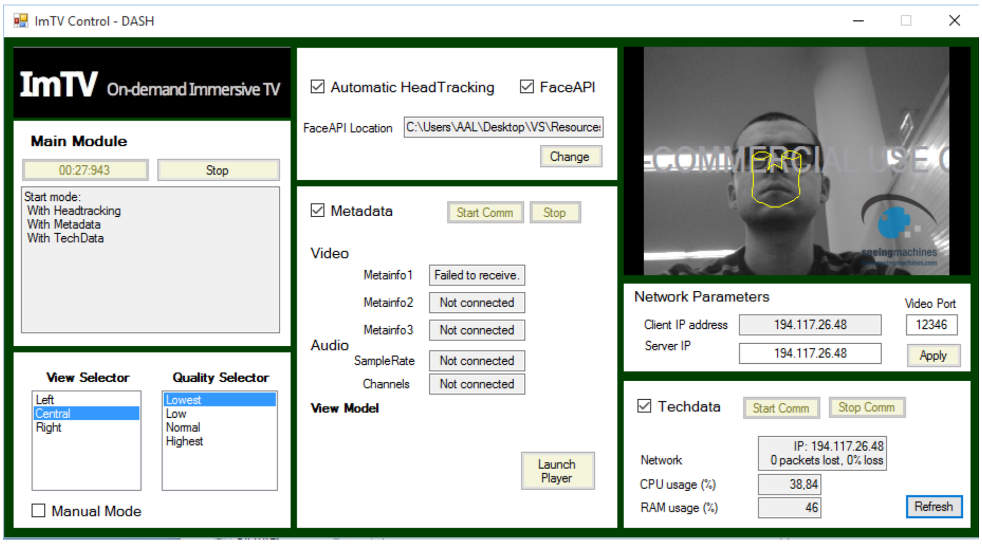

Figure 4.4: ImTV Version 1.

Both the MP4Client and the Client application are prepared for scalability, allowing the use of scenes with up to 5 different views and 7 different quality modes. They also support different view sets: *linear*, *cross* and *cardinal*.

Problem: In version 1, some of the features of the client interface are underdeveloped, resulting in a confusing and troubled user experience. The identified problems were:

1. The application must connect with the player in automatic mode in order to work properly. If the connection is established in manual mode, the view selection doesn't work.
2. Manual mode can only be activated after the connection in automatic mode is established.
3. Enabling manual mode closes FaceAPI.
4. The default view quality is "Lowest".
5. *Metadata* and *Techdata* panels are deactivated after the connection is established and do not display any information.
6. The *Techdata* information can only be displayed after pressing a "Refresh" button.
7. Some parameters have unclear or confusing designations.

Head tracking

As previously described, the client application uses a PlayStation Eye camera and FaceAPI for detecting the user's head movements and make a request to the player to change views.

Problem: The head tracking detection is poor and head tracking latency is very high.

Techdata

The client is prepared to read performance and networking information, including the number of packets lost, the percentage of packet loss and the CPU and RAM usage of the client.

Problem: The functionalities for reading the networking information are defective or are not sufficient for accurately monitoring bandwidth fluctuations.

Metadata

The client expects to read metadata about the video and audio components of the media stream, including the bitrate, sample rate, and the number of channels.

Problem: In Version 1, these functionalities are not fully implemented.

4.2.2 Problems and Underdevelopments

A priority was assigned to each problem as described in section 3.2.1. A summary of the identified problems is presented in table 4.2.

4.3 Version 2

4.3.1 Architecture

Figure 4.6 shows the architecture of version 2.

Table 4.2: Problems

Problem	Origin	Priority
The prototype does not support HTTP streaming.	MP4Box and MP4Client	High
Underdeveloped or confusing UI.	Client Application	Low
The head tracking detection is poor and head tracking latency is very high.	FaceAPI	Medium
Techdata information underdeveloped or insufficient.	Client Application	High
Metadata communication not fully implemented.	MP4Client	High

4.3.2 Changes and Improvements

The majority of the problems identified on version 1 are the result of an underdeveloped or faulty interaction between the different modules of the prototype.

In order to provide a more clear description of the developed solutions, this section addresses each solution and new development individually instead of describing the changes made in each module separately.

4.3.2.1 Support for Streaming

The default representation switching capabilities of the player are not compatible with multiview video. By default, the player automatically adapts the content to the bandwidth availability (e.g., in the default mode, when switching the quality of a central view representation, the player selects a right view instead of central view if this view occupies a lower bandwidth), preventing a proper interaction with the prototype.

The solution to this problem consists on overwriting the bandwidth parameter in the MPD and on deactivating automatic switching by adding to the DASH section of the GPAC configuration file the following parameter:

```
DisableSwitching=yes
```

This parameter deactivates all automatic switching, making impossible to use the default automatic adaptation of video quality. As such, a new adaptation scheme must be designed, where the client application should control all the switching operations. This modification along with the modifications described in chapter 5 and section 6.1, allows the prototype to support seamless streaming.

On the other hand, although the GPAC framework provides the DashCast tool for live/non-live streaming via MPEG-DASH [81], the prototype cannot support live-streaming. Previous tests show that the use of DashCast for live streaming (e.g., a surveillance application) is unfit for three reasons:

1. Only one representation is generated when the MPD is created.
2. The MPD is generated in real time and is constantly renewed.
3. The real-time coding of HD video samples also introduces an overload on the server and contributes to an increase in overall latency.

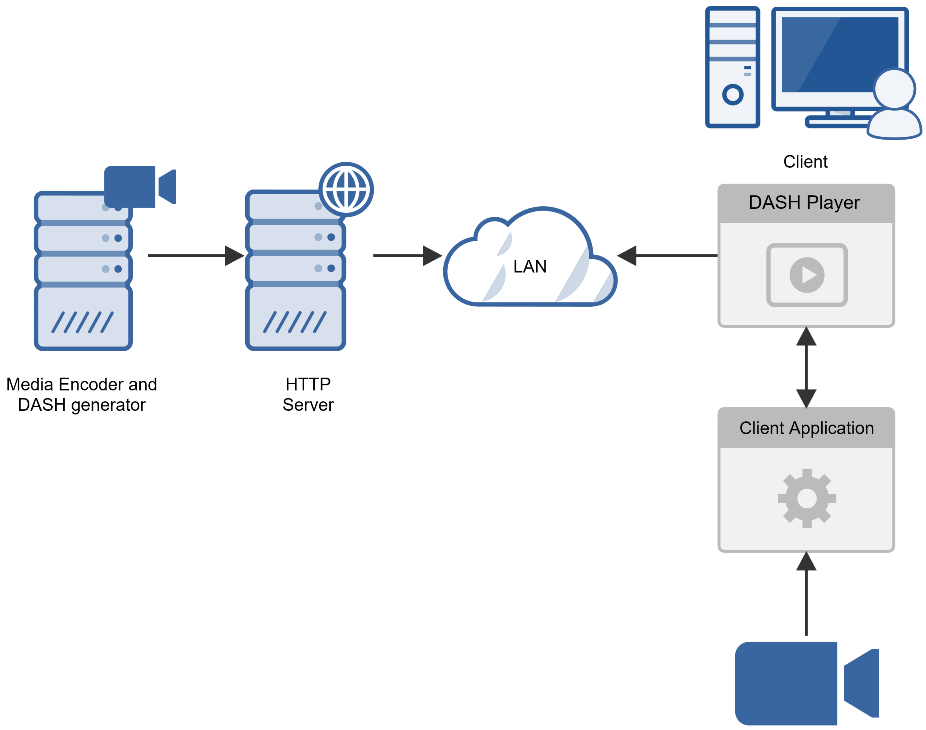

Figure 4.5: Version 2 Architecture.

As previously described, the prototype must override some of the MPEG-DASH specifications and requires constant support for several different representations, where a single MPD must be available at all times. Also, minimal latency is required in order to guarantee an immersive and interactive experience.

For these reasons, DashCast was dropped and instead MP4Box is used to generate the MPD and media segments. The generation of the MPD is fully addressed in chapter 5 and 6.

4.3.2.2 Client Application UI

The following changes were made to the Client Application UI:

1. Stopping the connection automatically turns off manual mode. Default (automatic mode) is restored.
2. For a better UX, enabling manual mode disables head tracking instead of closing FaceAPI. It's possible to toggle between manual and automatic mode, without closing FaceAPI.
3. The default mode is automatic. If the connection is established in manual mode, the application will automatically switch to automatic mode.
4. It's only possible to change to manual mode after the connection is established.

5. Upon connection, the application sets *central view* on *medium* quality as default.

6. Metadata information is displayed and updated automatically every second.

7. Techdata information is displayed and updated automatically every second. The refresh button has been removed.

8. In order to provide a more clear UI, some parameters designations were changed.

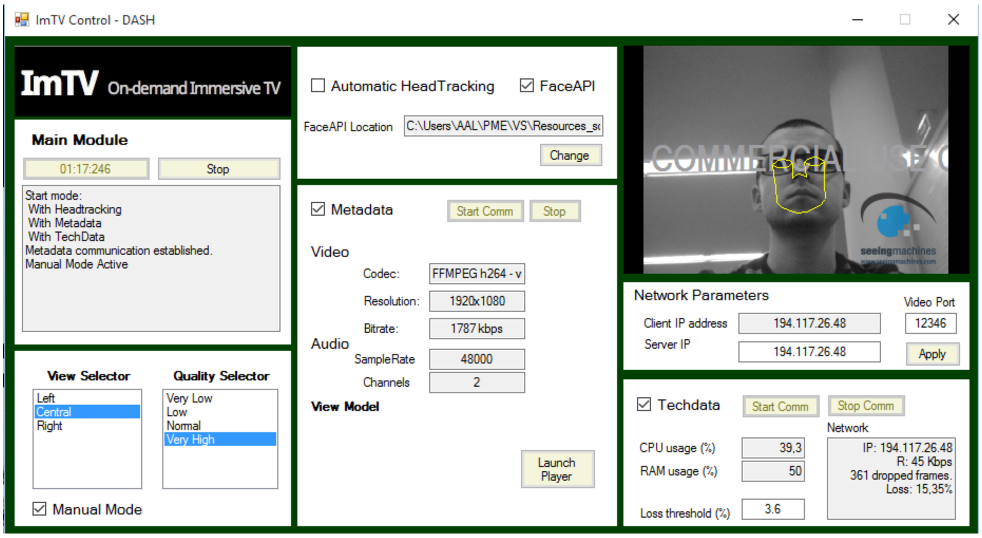

Figure 4.6: ImTV Version 2.

4.3.2.3 Head tracking latency

The only way to improve head tracking detection and to reduce the head tracking latency is with the use of a new camera. An F200 Intel RealSense Camera, with Intel RealSense SDK [82], was acquired. This camera promises better performance, lower latency and new forms of interaction. Due to time and availability constraints, it was not possible to test this camera in the prototype. All the tests and results described in this dissertation use the PlayStation Eye camera and FaceAPI. However, the F200 shows promising results and integration in the prototype is underway.

4.3.2.4 Techdata and Networking information

The client application was updated in order to take advantage of the features provided by the GPAC framework. Instead of directly computing the packet loss, the client application uses information received from the player to evaluate the QoS.

The bandwidth availability is measured in the client application by reading the global memory shared by the network card to compute the network utilization.

This information, alongside with the RAM and CPU usage of the client is displayed in the UI. The networking information, which is used in the automatic adaptation of video quality, is further explored in chapter 5.

4.3.2.5 Communication Protocols and Metadata

The user establishes the communication between the client application and the player. This communication provides the proper channels to send switching commands and to receive metadata information on the client application. The communication is built over two UDP connections. The following ports are used:

- **PORT 5010**: Switching operation communication.
- **PORT 5008**: Metadata communication.

On the switching operation protocol, once the connection is established, the player starts listening for any incoming messages. When the user stops the communication, the player and the client application close the connection. The metadata protocol is based on the switching operation. Figure 4.7 illustrates the Behaviour of the switching operation protocol, whereas 4.8 illustrates that of the metadata protocol.

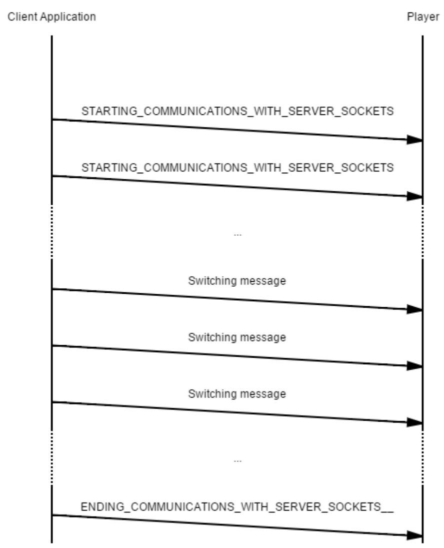

Figure 4.7: Switching operation communication protocol

The metadata protocol is based on the switching operation. The client application keeps retransmitting the starting message until the player replies with a metadata message. When the communication is stopped, the client sends the termination message to stop the communication and closes the connection. Upon receiving the termination message, the player immediately ceases the communication and closes the connection.

The metadata message structure is described in section 5.2.1.

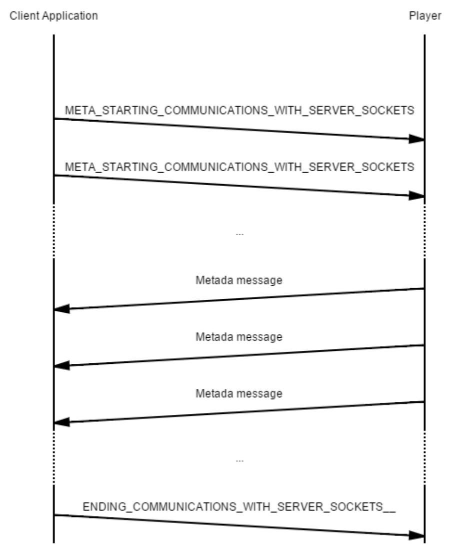

Figure 4.8: Metadata communication protocol

4.4 Player behaviour

Chapter 5 and 6 deal with the QoS and QoE. Before addressing these issues, it is important to fully comprehend the player behaviour. A thorough description is thus presented.

This description is a subset summary of the 'Example DASH client behaviour' present in Annex A of the ISO/IEC 230009-1 of a general DASH player [67], and includes the changes made to the player (i.e., mp4Client) to support the prototype requirements.

1. The player parses the MPD and selects the provided Adaptation Set;

2. Within the Adaptation Set, the player selects one representation based on the @*bandwidth* attribute. The bandwidth parameters are overwritten to ensure the bandwidth attributes do not interfere with the switching operations. The player automatically selects the representation with the lowest @*bandwidth* attribute (i.e., *central view* on *very low* quality), by default, as defined by the GPAC configuration file;

3. The players accesses the content by requesting segments or fragments of the media stream;

4. The client buffers the media for at least the value of *minimum buffer duration*, defined in the MPD, before starting the presentation. Once it identifies a Stream Access Point (SAP) for each representation, it starts rendering this SAP at wall-clock time (UTC). The representations are time-aligned to achieve seamless switching;

5. Once the presentation has started, the player continues to request media segments. The player can switch representations at the request of the client application. The player continues to check available bandwidth and adjust the buffer size during playback.

4.5 Summary

A summary of the state of the problems is presented in table 4.3.

Table 4.3: Problem status

Problem	Priority	State
The prototype does not support HTTP streaming.	High	Solved
Underdeveloped or confusing UI.	Low	Solved
The head tracking detection is poor and head tracking latency is very high.	Medium	Solved[1]
Techdata information underdeveloped or insufficient.	High	Solved
Metadata communication not fully implemented.	High	Solved

[1] Not tested.

Chapter 5

Quality of Service

It is important to guarantee quality of service to account for insufficient network capacity, especially in streaming situations. Even though QoS is the concern of CDNs, the prototype must guarantee that QoS is offered by adapting the video quality to network availability. This chapter addresses the issues related to the quality of service and performance of the prototype, namely, the reduction of switching latency and the adaptation algorithm.

5.1 Identification of Latency Sources and Reduced Latency.

The performance of the prototype is severely hindered by the switching latency and the head tracking latency. As a result, the latency in automatic mode is extremely high and severely damages the operability of the prototype, even when QoS is guaranteed. This section describes how the switching latency can be reduced to acceptable levels to ensure QoS is offered in several network scenarios. The head tracking latency is resolved by the use of a new camera, as described in section 4.3.2.3.

The player receives the messages from the client application and requests a representation switch in less than 1 s. Most of the delay occurs between the time the player makes the request to the time player displays a new representation. Without optimization, the bottleneck of this process (and of the prototype) is the switching operation itself. In this scenario, the switching latency can reach up to 6 s, when streaming from an HTTP server.

The causes for the delay introduced by this process are the following:

- Use of an inappropriate profile in the MPD.
- Poor optimization of the player capabilities.

The generation of an MPD with the appropriate profile and modifications on the GPAC configuration file allows to achieve a minimal delay on the switching operation.

5.1.1 MPD

5.1.1.1 Local Presentations

MP4Box allows the division of segments in sub-segments (or fragments). By default, in the *main* profile, the fragments have the same duration of the segments. The segments should be as small as possible in order to reduce the latency of the switching operation. For local presentations, the *main* profile must be used. The following MPD configuration is recommended and ensures a minimal latency in local presentations:

- Profile: Main
- Segment size: 1000 ms
- Fragment size: 50 ms
- Minimum buffer duration: 500 ms

The fragment size duration is close to the duration of an image (i.e., 40 ms). The player buffers media for 500 ms. The buffer has a length of 10 fragments.

5.1.1.2 Streaming from an HTTP server

When streaming from an HTTP server, the *main* profile increases the base switching latency in 2 s to 4 s. The utilization of the *On-Demand* profile offers a reduction of 2 s in streaming situations.

> 'This profile is intended to provide basic support for On-Demand content. The primary constraints imposed by this profile are the requirement that each Representation is provided as a single Segment, that Subsegments are aligned across Representations within an Adaptation Set and that Subsegments must begin with Stream Access Points. This permits a scalable and efficient use of HTTP servers and simplifies seamless switching." [67]

The following MPD configuration is used with the *On-Demand* profile:

- Profile: On-Demand
- Segment size: 120 770 ms
- Fragment size: 119 ms
- Minimum buffer duration: 100 ms

Every representation is divided into 1009 fragments with a duration of 119 ms. The fragments are time aligned and begin with SAP. The player buffer is filled with six fragments of every representation, creating a buffer with a depth of 714 ms for every representation.

5.1.2 GPAC Configuration

A relevant reduction of switching latency is achieved by altering some attributes in the GPAC configuration file. The following attributes were assigned in the *Systems* section:

```
Priority=real-time
ForceSingleClock=yes
```

With this configuration the system specifies 'real-time' priority to the decoders and a single timeline is used for all media streams [80].

The following attributes were assigned in the "DASH" section and are aligned with the *On-Demand* profile:

```
BufferMode=none
LowLatency=No
ImmediateSwitching=Yes
EnableBuffering=Yes
```

Buffering and immediate switching are enabled to account for streaming delay and to support seamless switching. The players settings for buffering are used (i.e, the player used the information provided by the MPD for buffering). The low-latency attribute is disabled because low latency modes can either increase the latency or make the presentation stall and may introduce artifacts [80]. This configuration ensures minimum switching latency.

5.2 Automatic Adaptation of Quality to Bandwidth Availability

The default control mechanism for adaptation of quality is disabled, as it does not support multiview streaming. This section describes a new mechanism for automatic adaptations of video quality to network conditions.

5.2.1 Metadata Communication

MP4Client has a built-in feature to provide information about the media streams. MP4Client was modified to collect and format this information and to support metadata communication. A *thread* is responsible for managing the metadata connection, as defined by the protocol described in 4.3.2.5.

- ***metadata_communication()***: *Thread* responsible for establishing the metadata connection with client application. This *thread* calls for the methods responsible for returning the metadata information. The metadata of the video and audio components are formatted into as a single string and sent to the client application. This process is repeated every second.

The following *thread* is responsible for managing the connection, on the client application side:

- ***MetadataCommunication()***: *Thread* responsible for establishing the metadata connection with player. This thread receives and parses the metadata message and provides a description of the media stream by updating the information displayed in the metadata panel and the value of parameters used in the adaptation scheme.

5.2.1.1 Metadata message structure

The metadata message has the following structure:

Table 5.1: Structure

Component	Video data			QoS data			Audio data		
Message structure	Code	Format	...	Code	Format	...	Code	Format	...

Table 5.2: Metadata message parameters

Component	Video data				QoS data		Audio data	
Parameter	Duration	Media codec	Resolution	Bitrate	Decoded frames	Dropped frames	Sample rate	Number of channels
Code	DUR	COD	RES	BIT	DEC	DRP	SMP	CHN
Format (Data type)	hh:mm:ss:ms (string)	- (string)	width x height (int)x(int)	- (int)	- (int)	- (int)	- (int)	- (int)

The metadata panel displays the *Media Codec*, *Resolution* and *Bitrate* parameters of the video component. The *Duration* parameter is not displayed or used by the client application. This panel also displays the *Sample Rate* and *Number of Channels* of the audio component. The QoS data is used in the adaptation scheme and the *Dropped frames* parameter is displayed in the *Techdata* panel of the client application.

There are two types of metadata messages:

- **Short metadata message**: Message sent every second. This message contains the bitrate, the number of lost frames and the number of decoded frames. The remaining fields are empty.

- **Long metadata message**: Message sent at the beginning of the communication or when a switching operation is requested. There are no empty fields. A variable detects if a switching operation occurs and stores the value assigned to the respective operation (the switching operation is thoroughly described in chapter 6). Every time this value changes, a flag signals the thread that a *short metadata message* should be sent.

The following methods are responsible for collecting and partially formatting the metadata:

- ***read_metadata()**: Method responsible for collecting and formatting the metadata. This method returns either the metadata of the video component plus the QoS data (i.e., number of dropped and decoded frames) or the metadata of the audio component.
- ***get_duration()**: Method responsible for formatting the media duration. Returns a *string* of the duration in the *hh:mm:ss:ms* format.
- ***get_number_dropped()***: Method responsible for collecting the number of lost frames. Returns the number of lost frames.

Disabled features

In order to minimize the number of messages sent it was considered to send short metadata strings only when a frame loss is detected. This approach was dropped for the following reasons:

1. Lack of support for a real-time evaluation of the player performance.
2. Poorer automatic adaptation of quality.
3. Load introduced by the metadata communication is minimal and can be neglected.

Nonetheless, this feature is implemented and *metadata_communication()* is prepared to send short metadata messages in this scenario.

5.2.2 Adaptation Algorithm

The adaption of quality is controlled by the client application. The following methods are responsible for the communication with the player, for displaying the *techdata* information and implementing the adaption algorithm:

- ***TechdataCommunication()***: *Thread* responsible for computing and updating the parameters used in the adaptation algorithm and the information displayed in the *techdata pannel*. The information is updated every second.
- ***HeadtrackingCommunication()***: *Thread* responsible for establishing the switching operation connection with player. The communication is unilateral. This thread processes and sends to the player the quality and view switching commands in automatic and manual mode and processes the adaptation algorithm described in this section.
- ***getNetworkUtilization()***: *Method* responsible for computing the networking information.
- ***ChangeActiveIndexinQualityBox()***: *Method* responsible for performing the quality switches.

The adaptation algorithm prioritizes seamless streaming over perceived video quality, as a determinant factor to provide superior QoE. The client applications reads and parses the messages received from the player and uses the bitrate, *QoS data* and the networking information in the adaption scheme. The following parameters define the quality of the streaming and the network conditions:

- ***videoBitrate***: The average bitrate over the last second.
- ***decodedFrames***: The current number of decoded frames.
- ***droppedFrames***: The current number of dropped frames.
- ***perLoss***: The percentage of lost frames.
- ***lossThreshold***: This parameter can be changed by the user and defines the threshold of the percentage of lost frames. The default value is 2.86%. Chapter 7 describes how this threshold value was computed.
- ***networkUtilization***: The current load of the network.

These parameters are displayed on the client application to allow the user to monitor the behavior of the player and the QoS.

The client application uses the *QoS data* to account for frame losses, originated by dropped packets, network errors, jitter or out of order delivery of packets. It is possible to measure the quality of the connection, by computing the percentage of lost frames.

The client application computes the load of the network and directly infers the bandwidth utilization by reading the network card memory utilization, to account for low-throughput situations due to network congestion.

An interval is defined using the *lossThreshold* as follows:

$$lossThreshold \times 0.54 \leq perLoss \leq lossThreshold \times 1.46 \tag{5.1}$$

For the network load the following interval is defined:

$$networkUtilization \times 0.34 \leq videoBitrate \leq networkUtilization \times 0.85 \tag{5.2}$$

The algorithm follows these simple rules:

1. If the *perLoss* and the *videoBitrate* parameters are inferior to the lower limit. The quality increases.
2. If the *perLoss* or the *videoBitrate* parameters are superior to the upper limit. The quality decreases.
3. If the *perLoss* and/or the *videoBitrate* lie between the interval limits. The quality does not change.

The algorithm is straightforward and supports scalability for an indefinite number of quality modes. It also ensures that no more than one switching operation occurs every second. In order to deliver a seamless adaptation of quality and offer better QoE, the adaptation is progressive (i.e., the algorithm only switches to adjacent quality modes). If the operation does not deliver the expected results the algorithm proceeds to perform a new switch. This process is repeated until the quality parameters are met.

5.3 Summary

Low switching latency is achieved by using the *On-Demand* profile and by altering some attributes in the GPAC configurations file. The QoS is only guaranteed if the *On-Demand* profile is used and the GPAC configuration file is correctly configured. For local presentations, the *main profile* should be used with segments with the duration of 1s and fragments with the duration of 50 ms.

With the optimizations described in this chapter is was possible to reduce the switching latency, when streaming from an HTTP server to 1.33 s; in the worst case scenario. The switching operation is no longer the bottleneck of the prototype. Chapter 7 presents the latency results obtained for different network conditions.

The quality adaptation algorithm proved to be effective and showed satisfying results. The intervals, as well the default loss threshold value presented above are defined as a result of tests performed, and further described in chapter 7.

Chapter 6

Quality of Experience

The prototype must offer good QoE to meet human expectations and deliver a seamless, natural and immersive experience. The several improvements and changes made to the prototype, already described in chapters 4 and 5 have an indirect effect on improving the overall experience. As such, this chapter addresses the major issue directly affecting the QoE, specifically, the transition of views.

6.1 Seamless Transition of Views

The player manages all the switching operations requested by the client. Two *threads* are responsible for managing the connection with the client and the switching operation. A switching operation refers to any switching between different representations (i.e., view or quality switching).

- ***data_reception()***: *Thread* responsible for establishing the switching operation connection with client application. The communication is unilateral, the *thread* parses the commands received by the client application and informs *player()* of the value of the switching operation to perform.

  Every command has a single digit value assigned, from 1 to 9, as depicted in Table 6.1. Codes 4 a 5 are not used in the prototype and are defined for scalability support. A flag establishes and signals the player if a switching operation is underway.

- ***player():*** *Thread* responsible for establishing the connection with the sever, reading and parsing the MPD, fetching the segments and managing the switching operations. Upon a request by the client, the client performs the switching operation, according to the code set by *data_reception()*.

On the client application the following thread is responsible for all the switching operation processing on the client applications side:

- ***HeadtrackingCommunication()***: *Thread* responsible for establishing the switching operation connection with player. The communication is unilateral. This thread processes and sends to the player the quality and view switching commands both in automatic and manual mode. On automatic mode, the head tracking information is provided by FaceAPI and used to determine the view switching operation to be performed.

Table 6.1: Command Code

Switching Operation	Representation	Command Code	Value	Type
View Switching	Central	CENTRAL_VIEW_	1	Standard
	Left	LEFT_1_VIEW__	2	Standard
	Right	RIGHT_1_VIEW_	3	Standard
	Very Left	LEFT_N_VIEW__	4	Scalability
	Very Right	RIGHT_N_VIEW_	5	Scalability
Quality Switching	Very Low	LOWER_QUALITY_	6	Standard
	Low	LOW_1_QUALITY_	7	Standard
	Medium	MIDDLE_QUALITY	8	Standard
	High	HIGH_1_QUALITY	-	Scalability
	Very High	HIGHER_QUALITY	9	Standard

6.1.1 Switching Operation

The MPD contains a single Period with a duration equal to the duration of the Media Streams (201 s) and a single Adaptation Set. Every representation is provided as a single segment. There are 12 representations, describing the 12 media streams. Every representation encompasses a video component and an audio component. In version 1 the adaptation set is presented as shown in Table 6.2.

Table 6.2: Adaptation Set - Version 1

Set of Views	Media Stream	Representation ID
Central	CenterPair verylow_dash.mp4	1
	CenterPair low_dash.mp4	2
	CenterPair medium_dash.mp4	3
	CenterPair high_dash.mp4	4
Right	RightPair verylow_dash.mp4	5
	RightPair low_dash.mp4	6
	RightPair medium_dash.mp4	7
	RightPair high_dash.mp4	8
Left	LeftPair verylow_dash.mp4	9
	LeftPair low_dash.mp4	10
	LeftPair medium_dash.mp4	11
	LeftPair high_dash.mp4	12
	teste_init.mp4	Initialization Segment

The switching operation is performed by the method ***gf_term_switch_quality()***. By manipulating two of the method parameters, it is possible to switch between different representations until the appropriate representation is selected. These parameters assume the following definition:

- ***direction***: Binary parameter assumes the value of 0 (up) or 1 (down). This parameter establishes the direction of the switching operation.

- ***number***: This parameter assumes an integer value and establishes the number of switching operations to be made in order to obtain the required representation.

In version 1, the 12 representations are grouped by view, composing 3 sets of 4 representations of media streams of different qualities. Every view switching operation to an adjacent view requires 4 switching operations. The number of switching operations S increases in multiples of 4 for non-adjacent representations:

$$S = 4n,\ n \in N \tag{6.1}$$

S increases in multiples of one, in quality switching operations of non-adjacent representations.

$$S = n,\ n \in N \tag{6.2}$$

This disparity in S for the two switching operations was identified as one of the sources for the video artifacts generated during a switching operation. The operation forces the player to switch between a large number of representations at run time while fetching the subsequent segments. The player has trouble finding an access point of the proper representation, failing to fetch some of the segments, causing loss of data.

In the quality switching operation, this phenomenon is not noticeable.

In order to eradicate this phenomenon, a different arrangement for the adaptation set was proposed. In version 2, the representations are grouped by quality, composing 4 sets of 3 representations. The MPD was generated as described in section 5.1.1.2. Every set encompasses a representation of each view in the same quality, as shown in Table 6.3.

Table 6.3: Adaptation Set - Version 2

Set of Views	Media Stream	Representation ID
Very Low	CenterPair verylow_dashinit.mp4	1
	LeftPair verylow_dashinit.mp4	2
	RightPair verylow_dashinit.mp4	3
Low	CenterPair low_dashinit.mp4	4
	LeftPair low_dashinit.mp4	5
	RightPair low_dashinit.mp4	6
Medium	CenterPair medium_dashinit.mp4	7
	LeftPair medium_dashinit.mp4	8
	RightPair medium_dashinit.mp4	9
Very High	CenterPair high_dashinit.mp4	10
	LeftPair high_dashinit.mp4	11
	RightPair high_dashinit.mp4	12
	prototype_init.mp4	Initialization Segment

In a first approach, it was considered to establish the left view as the first representation in each set in order to minimize S to the lowest value possible. In this scenario, the player starts the streaming from the central view and only a single switching operation for adjacent left and right view switching is required. This approach was be dropped because it was not feasible to start fetching segments from a middle representation in the set of views (i.e, the second representation in a 3-view set, the third in a 5-view set, the fifth in a 7-view set, etc.).

Instead, the central views are established as the first representations in each set, followed by alternate left and right views. This allows the player to automatically start fetching segments from the default central view.

With this arrangement, as depicted in Table 6.3, switching to the left view requires 1 switching operation and to the right view 2 switching operations. In scalability situations, on left and non-adjacent switching operations S is an odd number.

$$S = 2n+1,\ n \in N \tag{6.3}$$

On right switching, S is an even number.

$$S = 2n,\ n \in N \tag{6.4}$$

In quality switching, S increases in multiples of 3.

$$S = 3n,\ n \in N \tag{6.5}$$

Although this approach sacrifices S in quality switching operations, the trade-off greatly reduces the number of view switching operations and minimizes the maximum S, overall.

6.1.2 MPD

In order to achieve a seamless switching, in presentations from the local storage, it is necessary to provide as much access points as possible, without compromising the size of the segments, for a short time window. The MPD configuration described in section 5.1.1.1 is used.

When streaming from an HTTP-Server, the MPD configuration described in section 5.1.1.2 is used.

In both cases, it is defined that every fragment must begin with a stream access point (SAP), at the expense of sacrificing the fragments fixed length, if necessary.

With this addition to the MPD, it was possible to fully erase the artifacts and to provide seamless streaming during the switching operation.

6.2 Summary

The different arrangement used for the adaptation set reduces the total number of switching operations, S, overall, improves the quality of the switching operation and reduces the switching latency. By forcing the use of an SAP in every fragment, the remaining artifacts introduced, when switching views, are removed. The switching latency is also slightly reduced.

These modifications guarantee a seamless switching and a reduced number of switching operations (relevant in scalability situations where a large number of views or quality modes is used).

Chapter 7

Tests and Experiments

This chapter describes the experiences and tests performed and the results obtained. A description of the test scenarios can be found in section 3.3.

7.1 Performance and QoS

7.1.1 Simulation 1-A: Expected behaviour of the quality adaptation algorithm in a common streaming service in real world environments

This simulation assumes ideal network conditions (i.e,: maximum throughput and a frame loss of 0%). It is considered a hypothetical scenario where the algorithm only takes in account the network utilization values to do the adaptation.

7.1.1.1 Reference Values

Bandwidth

The average Internet connection speeds of different regions of the world can be found on the highlights of "Akamai's Q2 2015 State of the Internet" report [83] [84] [85]. The following reference values were chosen:

- 1.5 Mb/s: Paraguay (lowest average of the world);
- 5.1 Mb/s: World average;
- 23.1 Mb/s: South Korea (highest average of the world);
- 6.3 Mb/s: Turkey (lowest average in Europe);
- 6.459 Mb/s: Europe average;
- 10.4 Mb/s: Portugal;
- 16.1 Mb/s: Sweden (highest average in Europe).

Bitrate

Video streaming consumes a large share of bandwidth. It is expected that more than 80% of all consumer Internet traffic will be dominated by video streaming by 2019 [86]. As of late 2014, Youtube and Netflix, two of the most popular video streaming services accounted for 48.93 % of all downstream traffic during peak hours using fixed connection; in North America. Netflix alone accounts for 34.89% of this traffic and Youtube for 14.04%. Other video services like Amazon Video and Hulu amount to 2.58% and 1.41%, respectively. Together, these services amount to 52,92 % of all North American traffic [17] [87].

Netflix and Youtube were chosen as examples of common video streaming services. In this simulation, they provide a source of average bitrate values used in an ordinary video streaming service.

A correlation is found between the resolution and bitrate offered by Netflix and the resolution and recommended bitrate for YouTube videos coded in H.264 at 24 to 30 f/s. The two services offer slightly different resolutions with different aspect ratios for lower bitrates. [88] [89]. For simplicity, these different options are grouped into standardized types and the highest bitrate of each group is chosen. Table 1 shows this correlation 7.1.

Table 7.1: Expected bitrate of a common video streaming service

	Bitrate (kb/s)		
Type	Netflix (24/30 f/s)	YouTube (24/25/30 f/s)	WAM
240 p	235	N/A	235
360 p	750	1000	875
480 p (SD)	1750	2500	2125
720 p (HD)	3000	5000	4000
1080 p (HD)	5800	8000	6900
2160 p (4k)	N/A	35 000	35 000

This correlation represents the expected bitrate for different types and can be found by calculating the weighted arithmetic mean (WAM) of the bitrate values of the two services. These services do not offer a recommendation of bitrate value for 3D video. It is assumed the bitrate for 3D is close to the bitrate values of the HD (720 p and 1080 p) types. There are no YouTube recommendations for 240 p video [89] and Netflix uses HEVC to code UHD video [90]. Netflix recommend a bandwidth of 25 Mb/s for UHD video [91].

7.1.1.2 Quality adaptation algorithm behaviour

Table 7.2 shows the network utilization limits. The bandwidth values are multiplied by the network utilization limits. These limits are used in the adaptation algorithm and were adjusted according to the algorithm behaviour in the different scenarios. The quality adaptation algorithm is described in section 5.2.2. Figure 7.1 shows the behaviour of the quality adaptation algorithm in a streaming service.

Table 7.2: Network utilization limits

Region	Bitrate (kb/s)						
	World			Europe			
	Paraguay	World Avg.	S. Korea	Turkey	Europe Avg.	Portugal	Sweden
Avg. Bandwidth (kb/s)	1500	5100	23 100	6300	6459	10 400	16 100
Min. Network Utilization (kb/s)	510	1734	7854	2142	2196	3536	5474
Max. Network Utilization (kb/s)	1275	4335	19 635	5355	5490	8840	13 685

The lower network utilization limit was assigned to ensure that the adaptation algorithm guarantees, in the worst case scenario, QoS of the minimal quality in every region in the world. The upper network utilization limit was assigned to ensure that the HD (720 p) is supported in the largest number of regions possible. The limits were assigned as follows:

- Lower limit: $0.34 \times Networkutilization$
- Upper limit: $0.85 \times Networkutilization$

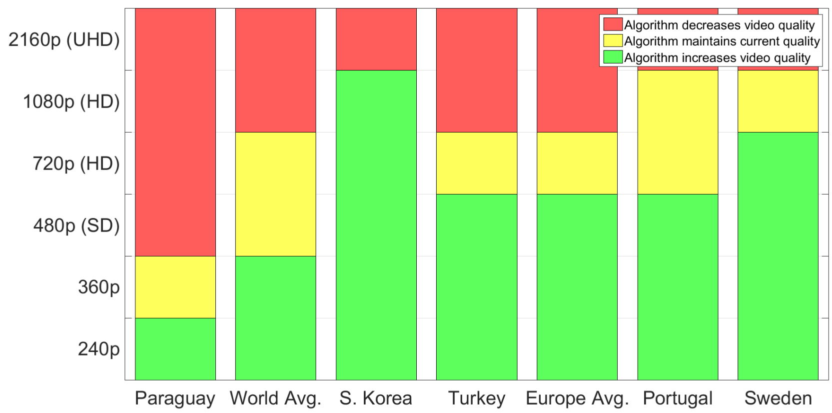

Figure 7.1: Expected behaviour of the quality adaption algorithm in a common streaming service

7.1.1.3 Congestion Limits

Using the reference value of 720 p, the congestion limits presented in Table 7.3 can be found. 720 p represents the highest quality mode that can be offered in the majority of the scenarios.

For every congestion limit, the adaption algorithm ceases to be able to offer a quality mode.

The lower limit represents a congestion close to ideal in which the algorithm can offer all qualities modes.

The upper limit represents the maximum congestion at which the algorithm can still offer all quality modes (up to 720 p).

Table 7.4 shows the expected throughput values at which all quality types are supported in different regions of the world. This table is built under the assumption that if HD (720 p) is

Table 7.3: Network congestion limits of a HD (720 p) video sample

Congestion limits (%)
1.00
22.00
37.00
39.00
62.00
76.00

supported, all the inferior quality types are also supported. The presented values follow the color scheme:

- Red: Does not support all quality types (up to 720 p);
- White: Supports all quality types (up to 720 p).

Table 7.4: Expected constrains on the support of HD (720 p) streaming

		Throughput (kb/s)						
Region		World			Europe			
		Paraguay	World Avg.	S. Korea	Turkey	Europe Avg.	Portugal	Sweden
Avg. Bandwidth (kb/s)		1500	5100	23 100	6300	6459	10 400	16 100
Bandwidth Availability (%)	99.00	1485	5049	22 869	6237	6394,41	10 296	15 939
	78.00	1170	3978	18 018	4914	5038,02	8112	12 558
	63.00	945	3213	14 553	3969	4069,17	6552	10 143
	61.00	915	3111	14 091	3843	3939,99	6344	9821
	38.00	570	1938	8778	2394	2454,42	3952	6118
	24.00	360	1224	5544	1512	1550,16	2496	3864

7.1.2 Simulation 1-B: Expected behaviour of the of the quality adaptation algorithm in the prototype in real world environments.

This simulation assumes ideal network conditions (i.e,: maximum throughput and a frame loss of 0%). It is considered a hypothetical scenario where the algorithm only takes in account the network utilization values to do the adaptation.

7.1.2.1 Reference Values

Bandwidth

The following bandwidth reference values were chosen based in the behaviour of the prototype in the simulation 1-A. These values were adjusted to ensure that, as the bandwidth increases, at least, one additional quality type is offered.

- Very Low: 1.5 Mb/s
- Low: 3.4 Mb/s

- Medium: 5.1 Mb/s
- High (e.g.: Europe): 6.4 Mb/s
- Very High (e.g.: Portugal): 10.4 Mb/s
- Unlimited (i.e., Maximum testbed throughput): 24.4 Mb/s

Bitrate

The video samples and bitrates used in this simulation can be found in section 4.2.1.1

7.1.2.2 Adaptation algorithm behaviour

Table 7.5 shows the expected behaviour of the prototype in real world environments.

Table 7.5: Network utilization limits in real world environments

	Bitrate (kb/s)					
Network capacity	Very Low	Low	Medium	High	Very High	Unlimited
Bandwidth (kb/s)	1500	3400	5100	6400	10 400	24 400
Min. Net. Utilization (kb/s)	510	1156	1734	2176	3536	8296
Max. Net. Utilization (kb/s)	1275	2890	4335	5440	8840	20 740

Figure 7.2 shows the behaviour of the quality adaptation algorithm in a streaming service.

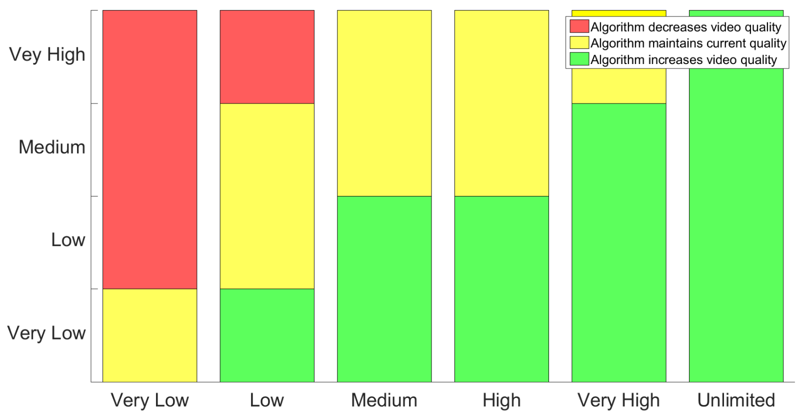

Figure 7.2: Expected behaviour of the prototype in real world environments

7.1.2.3 Congestion limits

Very high video quality is the highest quality mode available in the prototype and has a bitrate (3739 kb/s) close to the bitrate used in simulation 1-A (4000 kb/s). The congestion limits of the prototype can be found it Table 7.6.

Table 7.6: Network congestion limits for very high video quality

Congestion limits (%)
1.00
25.00
27.00
42.00
65.00

Table 7.7 shows the throughput values at which all the prototype quality modes are supported for different network capacities. The presented values follow the same color scheme of Simulation 1-A.

Table 7.7: Constrains on the support of the 'Very High' quality mode of the prototype

		Throughput (kb/s)					
Network capacity		Very Low	Low	Medium	High	Very High	Unlimited
Bandwidth (kb/s)		1500	3400	5100	6400	10 400	24 400
Bandwidth Availability (%)	99.00%	1485	3366	5049	6336	10 296	24 156
	75.00%	1125	2550	3825	4800	7800	18 300
	73.00%	1095	2482	3723	4672	7592	17 812
	58.00%	870	1972	2958	3712	6032	14 152
	35.00%	525	1190	1785	2240	3640	8540

7.1.3 Test 1: Performance evaluation of a presentation from local storage

7.1.3.1 Head tracking latency

The head tracking latency can be found in Table 7.8. The reference value represents the results of a test performed at INESC TEC, before this dissertation.

Table 7.8: Head tracking latency using Face API

Head movement	Surface	Reference	Avg. latency (s)
Center - Right	16:19:21-16:19:24	15:20:45-15:20:47	2.5
Center - Left	16:19:45-19:19:47	15:21:04-15:21:07	2.0
Left - Right	16:22:30-16: 22:32	15:21:43-15:21:45	2.0
Right - Center	16:22:36-16:22:38	15:22:25-15:22:27	2.0
Left - Center	16:24:14-16:24:16	15:23:00-15:23:03	2.5

7.1.3.2 Switching latency

Table 7.9 represent the total latency of view switching operation, ordered by the number of switching operations, *S*.

Table 7.10 represent the total latency of quality switching operation, ordered by the number of switching operations, *S*.

Table 7.9: View switching latency

S	Throughput		Surface	Desktop	Avg. latency(s)
1	Left - Right	Very Low	15:20:42-15:20:44	15:51:19-15:51:22	2.5
		Very High	15:19:59-15:20:01	15:38:27-15:38:30	2.5
1	Left - Center	Very Low	15:20:54-15:20:56	15:51:29-15:51:32	2.5
		Very High	15:20:11-15:20:13	15:38:37- 15:38:40	2.5
2	Center - Right	Very Low	15:21:00-15:21:02	15:51:36-15:51:39	2.5
		Very High	15:20:17-15:20:19	15:38:53-15:38:56	2.5

Table 7.10: Quality switching latency

S	Throughput	Surface	Desktop	Avg. latency(s)
3	Very Low - Low	15:18:55-15:18:58	15:37:50-15:37:53	3.0
3	Low - Medium	15:18:59-15:19:02	15:37:54-15:37:57	3.0
3	Medium - Very High	15:19:03-15:19:06	15:37:58-15:38:01	3.0
6	Very Low - Medium	15:19:19-15:19:22	15:38:05-15:38:08	3.0
6	Low - Very High	15:19:26-15:19:29	15:38:11-15:38:14	3.0
9	Very Low - Very High	15:19:47-15:19:50	15:38:17-15:38:20	3.0

7.1.3.3 Automatic mode latency

The quality switching latency values are:

- Minimum quality switching delay: 3.0 s
- Maximum quality switching delay: 3.0 s
- Average quality switching delay: 3.0 s

The view switching latency values using FaceAPI are:

- Minimum automatic switching delay: 4.5 s
- Maximum automatic switching delay: 5.0 s
- Average automatic switching delay: 4.7 s

7.1.4 Test 2: Performance evaluation when streaming from an HTTP server on a Client-Server Architecture using real-world specifications

7.1.4.1 Frame Loss

Using the bandwidth and congestion values identified in Experiment 1. Table 7.11 presents the result for the % loss in different scenarios. The presented values follow the color scheme:

- Red: The streaming stalls;

- White: Seamless streaming.

With unlimited network capacity (i.e., 24.4 Mb/s and 0% congestion) the average loss is of 0.26%.

Table 7.11: Frame loss in low throughput situations

Dropped packets (%) (Central-view)											
Video Quality		Very Low					Very High				
Network Capacity		Very Low	Low	Medium	High	Very High	Very Low	Low	Medium	High	Very High
Bandwidth (kb/s)		1500	3400	5100	6400	10 400	1500	3400	5100	6400	10 400
Network Congestion (%)	1.00	9.58	1.05	0.7	0.6	0.47	3.56	3.7	0.63	1.19	0.6
	25.00	2.76	0.95	0.39	0.14	0.35	6.27	2.65	6.92	1.46	0.94
	27.00	2.35	0.12	0.42	0.55	0.29	3.91	3.05	5.73	0.86	0.5
	42.00	1.56	0.18	0.6	0.42	0.24	5.13	2.22	3.06	5.84	0.61
	65.00	1.13	4.38	1.55	0.44	1.03	3.48	8.77	1.95	2.75	5.07

The following parameters represent the maximum, minimum and average loss in low-loss seamless streaming. The average loss was found by computing the weighted arithmetic mean (WAM) of the loss values.

- Minimum loss: 0.12 %
- Maximum loss: 1.55 %
- Average loss (WAM): 0.77 %
- Average number of lost frames: 38.77 (Total = 5035)

The performance values in in high-loss stalling streaming are:

- Minimum loss: 1.13 %
- Maximum loss: 9.58 %
- Average loss (WAM): 4.17 %
- Average number of lost frames: 210 (Total = 5035)

At 4.17% the streaming will stall. This value was identified as the maximum loss threshold (*Thmax*). For values above 1.55 %, the streaming may stall. This value was defined as the minimum loss threshold (*Thmin*).

The loss threshold (Th) is defined as follows:

$$Th = \frac{Thmax - Thmin}{2} + Thmin \tag{7.1}$$

Therefore, the threshold and the interval used in the adaptation algorithm described in 5.2.2 are defined as:

- Th: 2.86 %

- Thmax: $Th \times 1.46 = 4.17\%$
- Thmin: $Th \times 0.54 = 1.54\%$
- Range (Thmax-Thmin): 2.63%

7.1.4.2 Initial buffering duration

Table 7.12 presents the initial buffering duration, before starting the presentation.

Table 7.12: Initial buffering duration

Bandwidth(kb/s)		1500	3400	5100	6400	10 400	Avg. Time (s)
Congestion (%)	1.00	16:06:35-10:06:39	16:08:10-16:08:14	16:09:39-16:09:45	16:10:04-16:10:08	16:10:25-16:10:29	4.4
	65.00	16:17:35-16:17:39	16:17:10-16:17:17	16:13:45-16:13:52	16:14:06-16:14:12	16:14:26-16:14:31	5.8

7.1.4.3 Switching latency

Table 7.13 represent the total latency of the view switching operation, ordered by number of switching operations, *S*.

Table 7.13: View switching latency

S	Bandwidth (kb/s)		1500	6400	10 400	Avg. latency(s)
	Congestion (%)		1	1	0	
1	Left - Right	Very Low	14:28:58-14:28:58	14:41:21-14:41:21	15:05:07-15:05:07	0.00
		Very High	14:15:57-14:15:57	14:40:52-14:40:52	15:04:39-15:04:39	0.00
1	Left - Center	Very Low	14:29:06-14:29:06	14:41:17-14:41:17	15:05:12-15:05:12	0.00
		Very High	14:16:19-14:16:19	14:40:57-14:40:57	15:04:47-15:04:47	0.00
2	Center - Right	Very Low	14:29:15-14:29:15	14:41:25-14:41:25	15:05:16-15:05:16	0.00
		Very High	14:16:25-14:16:25	14:41:02-14:41:02	15:04:52-15:04:52	0.00

Table 7.14 represent the total latency of the quality switching operation, ordered by the number of switching operations, *S*.

Table 7.14: Quality switching latency

S	Bandwidth (kb/s)	1500	6400	10 400	Avg. latency(s)
	Congestion (%)	1	1	0	
3	Very Low - Low	14:14:57-14:14:58	14:39:55-14:39:57	15:03:51-15:03:52	1.33
3	Low - Medium	14:15:08-14:15:09	14:40:06-14:40:08	15:03:55-15:03:56	1.33
3	Medium - Very High	14:15:13-14:15:14	14:40:10-14:40:12	15:04:01-15:04:02	1.33
6	Very Low - Medium	14:15:24-14:15:25	14:40:22-14:40:23	15:04:12-15:04:13	1.33
6	Low - Very High	14:15:36-14:15:37	14:40:34-14:40:35	15:04:21-15:04:22	1.00
9	Very Low - Very High	14:15:47-14:15:48	14:40:44-14:40:45	15:04:27-15:04:28	1.00

7.1.4.4 Automatic mode latency

The quality switching latency values are:

- Minimum quality switching delay: 1.00 s
- Maximum quality switching delay: 1.33 s
- Average quality switching delay: 1.22 s

The view switching latency values using FaceAPI are:

- Minimum automatic switching delay: 2.00 s
- Maximum automatic switching delay: 2.50 s
- Average automatic switching delay: 2.20 s

7.2 Subjective tests and QoE

7.2.1 Test 1-A: QoE evaluation of a presentation from local storage

Manual Mode

- Seamless streaming: 5
- Perceived video quality: 5
- User friendly input: 4
- Seamless view switching: 2
- Seamless quality switching: 4

ACR: 4 (Good)

Automatic Mode

- Seamless streaming: 5
- Perceived video quality: 5
- User friendly input: 1
- Seamless view switching: 2
- Seamless quality switching: 4

ACR: 3.4 (Fair)

Local presentations offer an average QoE, severely hindered by the high view switching latency. The quality switching latency, despite its high value, has a diminished effect of the QoE. The presentation quality is very good. There's no initial buffering, so the presentation starts right away.

7.2.2 Test 1-B: Presentation from an HTTP server on a client-server architecture in low-loss situations

Manual Mode

- Seamless streaming: 5
- Perceived video quality: 5
- User friendly input: 4
- Seamless view switching: 5
- Seamless quality switching: 5

ACR: 4.8 (Excellent)

Automatic Mode

- Seamless streaming: 5
- Perceived video quality: 5
- User friendly input: 1
- Seamless view switching: 5
- Seamless quality switching: 5

ACR: 4.2 (Good)

Presentation and view switching of very-high-quality, severely hindered by poor head tracking detection and high head tracking latency. There are no noticeable loss or artifacts. The quality adaption algorithm has no noticeable negative side effect on the QoE, the adaptation is seamless and the adaptation time is appropriate.

7.2.3 Test 1-C: Presentation from an HTTP server on a client-server architecture in high-loss situations

Manual Mode

- Seamless streaming: 2
- Perceived video quality: 1
- User friendly input: 4
- Seamless view switching: 4
- Seamless quality switching: 4

ACR: 3.0 (Fair)

Automatic Mode

- Seamless streaming: 2
- Perceived video quality: 1
- User friendly input: 1
- Seamless view switching: 4
- Seamless quality switching: 4

ACR: 2.4 (Poor)

Presentation of poor quality, there's no noticeable loss or artifacts, but the presentation stalls often. Poor head tracking detection and high head tracking latency. View switching and quality show a slightly poorer performance due to presentation stalling. The quality adaption algorithm has no noticeable negative side effect on the QoE, the adaptation is immediate and the adaptation time is appropriate.

Chapter 8

Conclusions and Future Work

This chapter summarizes the results obtained. An overview of the satisfaction of the prototype objectives and the main conclusions is given. It finishes with a suggestion of issues, changes and improvements to look up in the future.

8.1 Results

All the dissertation objectives were met and a fully-functional FTV system was obtained. This system is applicable to a different range of scenarios of VOD streaming and is designed to provide an immersive experience for diverse network conditions. This system offers seamless streaming and the ability to explore 3D scenes from different viewpoints.

A quality adaptation algorithm was developed. This algorithm dynamically adapts the video quality to the bandwidth availability. The results suggest that the adaptation algorithm can be modified or used in a common video streaming service and on different types of video applications.

Additional research was performed on the behaviour of the adaptation algorithm and the applicability of the prototype to real world environments and a performance evaluation was performed.

In congestion situations, at a loss between 1.13% and 1.55%, the player behaviour is unstable and streaming may or may not stall. In these situations, the prototype appears to perform better if the loss is relatively constant, making seamless streaming possible in these situations. If the loss occurs in bursts, the prototype will manifest a poorer behaviour. Good QoE is guaranteed for losses bellow 1.13%. A loss above 4.17% can severely reduce the QoE. Good QoE is not guaranteed in these situations.

In this system, the initial buffering duration is independent of network capacity and increases with network congestion. This increase can be of 1.4 s in high congestion scenarios. With the current configurations, the initial buffer time is inferior or equal to 5.8 s.

The original switching latency of 6.0 s was reduced to 1.33 s in the worst case scenario, on quality switching. On view switching, it was reduced to 0.0 s. There's no noticeable latency and the switching is immediate.

The number of switching operations S has no noticeable effect on the switching latency, but has an effect on the perceived quality of the view switching operation. It is important to note that some frames may be lost during the switching operation. This loss has no noticeable effect on the performance of the prototype.

The results also suggest that the prototype is supported is most regions of the world.

In manual mode, the prototype offers an excellent QoE. In this mode, the bottleneck of the system is the switching operation. In automatic mode (the scope of this dissertation), the prototype offers a good QoE, severally hindered by the head tracking operation.

8.2 Conclusion

The current network capacities still fall behind on the recommend bandwidth requirements to support HD and 3D video, as both Europe's and the world's average network capacities can only support HD 720p. However, this problem can be solved by the use of HEVC, as video bitrate can be reduced in half while maintaining the same perceived quality. With HEVC, video streaming services can offer more quality types for regions in the world with lower network capacity. Adaptive schemes, as presented in this dissertation can also offer video streaming in high quality for lower bitrates. On the other hand, Portugal's network capacity is twice the world's average and can offer video streaming in HD effortlessly.

It remains a challenge how CDNs will be able to support the imminent UHD qualities as the vast majority of the networks cannot support the bitrate requirements. Even with the use of HEVC, UHD and 3D delivery remains a challenge.

DASH appears to be the ideal method for streaming content on the Internet. Using the *On-Demand* profile, this technique can offer seamless streaming with low latency and low loss, even for poor network conditions.

The automatic quality adaptation algorithm presented in this dissertation can be adjusted to any common streaming service and any video type, independently of the number of views, and is expected to offer good results.

The prototype can offer good QoE in different types of systems with at least 4 GiB of RAM and for different types of networks. The prototype is supported in most regions of the world, as the maximum bitrate used is inferior the world's average bandwidth. The maximum bitrate used in the prototype is enough to offer very good video quality of 3D multiview video.

The prototype is scalable and can be easily adapted to more views and unlimited quality modes, with no expected or little reduction on performance and QoE. The prototype is expected to effortlessly support any kind of 3-view set, as long as the MPD and media segments generation follow the recommendations described in this document. The prototype can also use other view modes.

It is expected, with the use of the new camera, that the prototype will be able to offer a similar and excellent QoE in manual and automatic mode.

8.3 Future Work

In order to reduce visual fatigue and to support a complete 3D view of a scene, it is suggested the capture scenes using 64 views [29]. In automatic mode, the head tracking operation severely deteriorates the overall experience. Efforts to reduce the latency introduced by this process are already underway and this problem is expected to be solved with the use of a new camera and API.

Other methods of interaction using new technologies are suggested. Further research on HMDs and mobile devices [92] is required to evaluate the feasibility of the adaptation of the prototype to these devices, but it is an interesting idea to use the motion detection (head and eye tracking) capabilities of these devices for automatic view switching.

The prototype cannot support live streaming using the GPAC framework, limiting the applicability of the prototype to a broader range of situations (e.g., video-surveillance). Additional research on new frameworks to support live streaming is necessary.

The prototype uses and older version of the GPAC framework, and cannot support media streams coded using HEVC. Most recent versions of GPAC already support streaming of HEVC via DASH [93]. It is suggested research on the feasibility of using the updated frameworks in the prototype in order to support and take advantage of HEVC coding capabilities. It is possible to achieve a reduction of more than 50% on the bitrate for the same quality, using 3D-HEVC [44].

GPAC can be embedded in Mozilla-based and Internet Explorer web browsers [94]. It is suggested the adaptation of the prototype to use in web browsers as well as additional research on new frameworks to support the integration on other browsers like Google Chrome or Microsoft Edge.

On the client side, the player and the client application run separately, creating a clunky and poor interface for the user. It is suggested the integration of the player in the client application.

The client application can be used to predict the users' preferences locally. This information can be used by the content providers to study the users' view preference using supervised learning algorithms and allow the client to automatically start the presentation with the users' favourite view set, reducing the number of switching operations, and improving the overall experience. SVNs, decision trees and neural networks are recommended to analyze this information.

The prototype offers a new type of immersive experience for on-Demand content. User's can effortlessly explore a 3D scene from different viewpoints. New kinds of media content can be offered with this type of system and content providers can effortlessly offer a new type of content without an extra load on the CDNs. Although, the capture of multiview content appears to be the most difficult task of the production stage, multiview video skips the video stitching stage of 360º video and does not require a special player. With 64-view, a full 3D scene can be offered. This type of configuration can be used in sports or musical events to provide and interactive and immersive experiences.

References

[1] S Kunic and Z Sego. 3D television. *Proceedings ELMAR-2011*, (April 1953):127–131, 2011. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6044313.

[2] Highest grossing 3D movies in North America 2015 | Statistic. URL: http://www.statista.com/statistics/348870/highest-grossing-3d-movies/ [last accessed 2015-10-09].

[3] The Illustrated 3D Movie List. URL: http://www.3dmovielist.com/list.html [last accessed 2015-10-20].

[4] Cinema - 3D screens worldwide 2014 | Statistic. URL: http://www.statista.com/statistics/271863/number-of-3d-cinema-screens-worldwide/ [last accessed 2015-10-09].

[5] Will 3D movies disappear from our screens? URL: http://moviepilot.com/posts/2015/02/07/will-3d-movies-disappear-from-our-screens-2669692?lt{_}source=external,manual [last accessed 2015-10-20].

[6] 4 Reasons Why 3D Films Have Failed. URL: https://www.brandwatch.com/2015/05/4-reasons-why-3d-films-have-failed/ [last accessed 2015-10-19].

[7] 3D Movies at the Box Office - Box Office Mojo. URL: http://www.boxofficemojo.com/genres/chart/?id=3d.htm [last accessed 2015-10-19].

[8] 3D Release Calendar — BoxOffice®. URL: http://pro.boxoffice.com/statistics/3d-release-calendar [last accessed 2015-10-20].

[9] The Decline of 3-D? Not So Fast - The New York Times. URL: http://www.nytimes.com/2015/01/08/arts/international/the-decline-of-3-d-not-so-fast.html?{_}r=0 [last accessed 2015-10-20].

[10] Virtual realiy HMD unit sales worldwide 2014-2018 | Statistic. URL: http://www.statista.com/statistics/426429/hmd-virtuial-reality-unit-sales-worldwide/ [last accessed 2015-10-20].

[11] Facebook launches 360-degree videos - and they're amazing. URL: http://money.cnn.com/2015/09/23/technology/facebook-360-video/ [last accessed 2015-10-09].

[12] You Can Now Watch and Upload 360-Degree Videos on YouTube | WIRED. URL: http://www.wired.com/2015/03/youtube-360-degree-video/ [last accessed 2015-10-09].

[13] The Growing Virtual/Augmented Reality Space: Funding Trends And Active Investors. URL: https://www.cbinsights.com/blog/virtual-augmented-reality-funding/ [last accessed 2015-10-20].

[14] Adoption of TV technologies in the U.S. 2014 | Statistic. URL: http://www.statista.com/statistics/198738/adoption-and-use-of-key-audio-visual-services-in-the-united-states/ [last accessed 2015-10-09].

[15] Monthly active Spotify users worldwide 2015 | Statistic. URL: http://www.statista.com/statistics/367739/spotify-global-mau/ [last accessed 2015-10-09].

[16] U.S. music streaming revenue 2014 | Statistic. URL: http://www.statista.com/statistics/437717/music-streaming-revenue-usa/ [last accessed 2015-10-09].

[17] Chart: Netflix and YouTube Are America's Biggest Traffic Hogs | Statista. URL: http://www.statista.com/chart/1620/top-10-traffic-hogs/ [last accessed 2015-10-09].

[18] Chart: The Internet Is Gradually Replacing Traditional Media | Statista. URL: http://www.statista.com/chart/3519/media-consumption/ [last accessed 2015-10-09].

[19] Music tracks volume on streaming sites 2015 | Statistic. URL: http://www.statista.com/statistics/435256/number-tracks-on-music-streaming-services/ [last accessed 2015-10-09].

[20] Amazon Prime vs Netflix - Difference and Comparison | Diffen. URL: http://www.diffen.com/difference/Amazon{_}Prime{_}Instant{_}Video{_}vs{_}Netflix [last accessed 2015-10-20].

[21] Sylvain Thiébaud Doyen, Didier. TOWARDS A FREE VIEWPOINT AND 3D INTENSITY ADJUSTMENT ON MULTI-VIEW DISPLAY. pages 81–85, 2014.

[22] DigitalMediaLab. Research_2D3DconversionThree types of binocular parallax.jpg. URL: http://image.kwangwoon.ac.kr/research/research{_}2D3Dconversion.htm [last accessed 2015-05-01].

[23] Maria Teresa Andrade. Televisão digital e novos serviços - 3D. *Televisão Digital e Novos Serviços*, 2015.

[24] Fernando Pereira. 3D Video Systems. *Video Systems*, 2012.

[25] Donghyun Kim, Sunghwan Choi, and Kwanghoon Sohn. Effect of vergence-accommodation conflict and parallax difference on binocular fusion for random dot stereogram. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(5):811–816, 2012. doi:10.1109/TCSVT.2012.2186738.

[26] Min-Chul Park and Sungchul Mun. Overview of Measurement Methods for Factors Affecting the Human Visual System in 3D Displays. *Journal of Display Technology*, (c):1–1, 2015. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7004777, doi:10.1109/JDT.2015.2389212.

[27] Jincheol Park, Sanghoon Lee, and Alan Conrad Bovik. 3D visual discomfort prediction: Vergence, foveation, and the physiological optics of accommodation. *IEEE Journal on Selected Topics in Signal Processing*, 8(3):415–427, 2014. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6767068, doi:10.1109/JSTSP.2014.2311885.

[28] Janusz Konrad and Michael Halle. 3-D displays and signal processing. *IEEE Signal Processing Magazine*, 24(6):97–111, 2007. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4388084, doi:10.1109/MSP.2007.905706.

[29] Hakan Urey, Kishore V. Chellappan, Erdem Erden, and Phil Surman. State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, 99(4):540–555, 2011. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp={&}arnumber=5701756, doi:10.1109/JPROC.2010.2098351.

[30] Steven M Lavalle, Anna Yershova, and Max Katsev. Head Tracking for the Oculus Rift. 2014. doi:10.1109/ICRA.2014.6906608.

[31] Engineering and Technology Magazine. TheGraphic. (March):2014, 2014. URL: http://eandt.theiet.org/news/2014/feb/rugby-oculus-rift.cfm.

[32] Wikipedia. Parallax_barrier_vs_lenticular_screen. URL: http://upload.wikimedia.org/wikipedia/commons/a/a8/Parallax{_}barrier{_}vs{_}lenticular{_}screen.svg.

[33] Gordon Wetzstein, Douglas Lanman, Matthew Hirsch, Wolfgang Heidrich, and Ramesh Raskar. Compressive light field displays. *IEEE Computer Graphics and Applications*, 32(5):6–11, 2012. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6311368, doi:10.1109/MCG.2012.99.

[34] Péter Tamás Kovács, Kristóf Lackner, Attila Barsi, Ákos Balázs, Atanas Boev, Robert Bregovi, and Atanas Gotchev. Measurement of perceived spatial resolution 3d light-field displays.measurement of perceived spatial resolution 3d light-field displays. pages 768–772, 2014. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=7025154, doi:10.1109/ICIP.2014.7025154.

[35] Yasuhiro Takaki. Super multi-view display and holographic display. *Conference Proceedings - Lasers and Electro-Optics Society Annual Meeting-LEOS*, 4:12–13, 2009. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5343495, doi:10.1109/LEOS.2009.5343495.

[36] Xuewu Xu, Yuechao Pan, Phyu Phyu Mar Yi Lwin, and Xinan Liang. 3D holographic display and its data transmission requirement. *2011 International Conference on Information Photonics and Optical Communications, IPOC 2011*, pages 1–4, 2011. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6122872, doi:10.1109/IPOC.2011.6122872.

[37] Ilmãrs Osmanis, Krišs, Valters, Gatis, Osmanis. 3D Volumetric Display Design Challenges. 2013. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=6702001, doi:10.1109/NORCHIP.2013.6702001.

[38] Y Wu and Y Zhao. Brief analysis of the H. 264 coding standard. *Iih-Msp*, 2007. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4457675, doi:10.1109/IIHMSP.2007.4457675.

[39] Ying Chen and Anthony Vetro. Next-generation 3D formats with depth map support. *IEEE Multimedia*, 21(2):90–94, 2014. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6818898, doi:10.1109/MMUL.2014.31.

[40] Patrick Seeling and Martin Reisslein. Video Traffic Characteristics of Modern Encoding Standards: H.264/AVC with SVC and MVC Extensions and H.265/HEVC. *The Scientific World Journal*, 2014:1–16, 2014. URL: http://www.hindawi.com/journals/tswj/2014/189481/, doi:10.1155/2014/189481.

[41] Philippe Hanhart, Naeem Ramzan, Vittorio Baroncini, and Touradj Ebrahimi. CROSS-LAB SUBJECTIVE EVALUATION OF THE MVC + D AND 3D-AVC 3D VIDEO CODING STANDARDS. pages 183–188, 2014. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp={&}arnumber=6982317, doi:10.1109/QoMEX.2014.6982317.

[42] Gary J. Sullivan, Jens Rainer Ohm, Woo Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6316136, doi:10.1109/TCSVT.2012.2221191.

[43] Ying Chen, Li Zhang, Vadim Seregin, and Ye-kui Wang. Motion Hooks for the Multiview Extension of HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(12):2090–2098, dec 2014. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=6887363, doi:10.1109/TCSVT.2014.2352571.

[44] Gabriel Balota, Mário Saldanha, Gustavo Sanchez, Bruno Zatt, Marcelo Porto, and Luciano Agostini. Overview and quality analysis in 3D-HEVC emergent video coding standard. *2014 IEEE 5th Latin American Symposium on Circuits and Systems, LASCAS 2014 - Conference Proceedings*, pages 0–3, 2014. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6820260, doi:10.1109/LASCAS.2014.6820260.

[45] M. Tanimoto. Overview of FTV (free-viewpoint television). *2009 IEEE International Conference on Multimedia and Expo*, 4:1552–1553, 2009. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=5202803, doi:10.1109/ICME.2009.5202803.

[46] Masayuki Tanimoto. Ftv standardization in mpeg. pages 4–7, 2014. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=6874767, doi:10.1109/3DTV.2014.6874767.

[47] Kai Chen Huang, Po Yu Chien, Cheng An Chien, Hsiu Cheng Chang, and Jiun In Guo. A 360-degree panoramic video system design. *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test, VLSI-DAT 2014*, pages 11–14, 2014. doi:10.1109/VLSI-DAT.2014.6834863.

[48] How 360 video works ? - 360 VR Video software | VideoStitch. URL: http://www.video-stitch.com/how-it-works/ [last accessed 2015-07-15].

[49] Spherical 360 Video | AirPano.com project. URL: http://www.airpano.com/360-videos.php [last accessed 2015-07-15].

[50] Panocam3D - 3d 360 videos. URL: http://www.panocam3d.com/video3d360.html [last accessed 2015-07-15].

[51] Live VR Video software | VideoStitch Vahana VR. URL: http://www.video-stitch.com/vahana-vr/ [last accessed 2015-07-15].

[52] Jacob Chakareski. Adaptive multiview video streaming: Challenges and opportunities. *IEEE Communications Magazine*, 51(5):94–100, 2013. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6515052, doi:10.1109/MCOM.2013.6515052.

[53] Olivier Hamidouche, Wassim Raulet, Mickael Déforges. REAL TIME SHVC DECODER : IMPLEMENTATION AND COMPLEXITY ANALYSIS. pages 2125–2129, 2014. URL: http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7025426, doi:10.1109/ICIP.2014.7025426.

[54] Iraj Sodagar, Anthony Vetro, and Iraj Sodagar. Industry and Standards The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67, 2011. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=6077864, doi:10.1109/MMUL.2011.71.

[55] Truong Cong Thang, Quang Dung Ho, Jung Won Kang, and Anh T. Pham. Adaptive streaming of audiovisual content using MPEG DASH. *IEEE Transactions on Consumer Electronics*, 58(1):78–85, 2012. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6170058, doi:10.1109/TCE.2012.6170058.

[56] Show, Cyril Concolato, and Jean Le Feuvre. Overhead and Performance of Low Latency Live Streaming using MPEG-DASH. *IISA QoE for Over-The-Top Multimedia Services*, 2014. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6878732, doi:10.1109/IISA.2014.6878732.

[57] Sunghee Lee, Kimyung Youn, Kwangsue Chung, and Senior Member. Adaptive Video Quality Control Scheme to Improve QoE of MPEG DASH. 1(4):126–127, 2015. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7066348, doi:10.1109/ICCE.2015.7066348.

[58] Tianyu Su, Abbas Javadtalab, Abdulsalam Yassine, and Shervin Shirmohammadi. A DASH-Based 3D Multi-view Video Rate Control System. pages 1–6, 2014. URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp={&}arnumber=7021071, doi:10.1109/ICSPCS.2014.7021071.

[59] Rob Schapire. Machine Learning Algorithms for Classification. Technical report, Princeton University.

[60] A Tour of Machine Learning Algorithms - Machine Learning Mastery. URL: http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ [last accessed 2015-06-28].

[61] Supervised Learning Workflow and Algorithms - MATLAB & Simulink. URL: http://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html [last accessed 2015-06-28].

[62] Comparing supervised learning algorithms. URL: http://www.dataschool.io/comparing-supervised-learning-algorithms/ [last accessed 2015-06-28].

[63] Support Vector Machines (SVM) - MATLAB & Simulink. URL: http://www.mathworks.com/help/stats/support-vector-machines-svm.html [last accessed 2015-06-28].

[64] Classification Trees and Regression Trees - MATLAB & Simulink. URL: http://www.mathworks.com/help/stats/classification-trees-and-regression-trees.html [last accessed 2015-06-28].

[65] Neuron Model - MATLAB & Simulink. URL: http://www.mathworks.com/help/nnet/ug/neuron-model.html [last accessed 2015-06-28].

[66] Neural Network Architectures - MATLAB & Simulink. URL: http://www.mathworks.com/help/nnet/ug/neural-network-architectures.html [last accessed 2015-06-29].

[67] INTERNATIONAL STANDARD ISO / IEC Information technology — Dynamic. 2014, 2014.

[68] About us | GPAC. URL: https://gpac.wp.mines-telecom.fr/home/about/ [last accessed 2015-10-14].

[69] Node.js. URL: https://nodejs.org/en/ [last accessed 2015-10-23].

[70] FaceApi. URL: http://www.seeingmachines.com/wp-content/uploads/2014/10/sm11069-faceAPI-Production-License-FINAL.pdf [last accessed 2016-01-05].

[71] ShareLaTeX, the Online LaTeX Editor. URL: https://www.sharelatex.com/ [last accessed 2015-06-30].

[72] The differences between 2.4GHz and 5GHz Wireless - Welcome to TP-LINK. URL: http://www.tp-link.com/en/faq-499.html [last accessed 2016-01-05].

[73] iPerf - The TCP, UDP and SCTP network bandwidth measurement tool. URL: https://iperf.fr/ [last accessed 2016-01-05].

[74] NetLimiter Homepage. URL: http://www.netlimiter.com/ [last accessed 2016-01-05].

[75] Itu-T. Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment. 2014. URL: https://www.itu.int/rec/T-REC-P.913-201401-I/en.

[76] FFmpeg. URL: https://www.ffmpeg.org/ [last accessed 2016-01-05].

[77] MP4Box | GPAC. URL: https://gpac.wp.mines-telecom.fr/mp4box/ [last accessed 2015-10-14].

[78] Features | GPAC. URL: https://gpac.wp.mines-telecom.fr/player/features/ [last accessed 2015-10-14].

[79] DASH Features in GPAC Players | GPAC. URL: https://gpac.wp.mines-telecom.fr/player/features/dash/ [last accessed 2015-10-14].

[80] Process Configuration - Documentation. URL: http://htmlpreview.github.io/?https://github.com/gpac/gpac/blob/master/doc/configuration.html [last accessed 2015-10-14].

[81] DashCast | GPAC. URL: https://gpac.wp.mines-telecom.fr/dashcast/ [last accessed 2015-10-14].

[82] RealSense - Developing for the Laptop F200 | Intel® Developer Zone. URL: https://software.intel.com/en-us/realsense/f200camera [last accessed 2016-01-05].

[83] Mobile Connectivity. Internet & broadband adoption - EMEA. (Q):6–8, 2014.

[84] Mobile Connectivity. Internet & broadband adoption - ASIA -PACIFIC. pages 6–8, 2014.

[85] Mobile Connectivity. Internet & broadband adoption - AMERICAS. pages 6–8, 2014.

[86] Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper. URL: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white{_}paper{_}c11-481360.html [last accessed 2015-11-30].

[87] Netflix now accounts for 35% of bandwidth usage in the US and Canada - Quartz. URL: http://qz.com/299989/netflix-now-accounts-for-35-of-bandwidth-usage-in-the-us-and-canada/ [last accessed 2015-11-30].

[88] The Netflix Tech Blog: Per-Title Encode Optimization. URL: http://techblog.netflix.com/2015/12/per-title-encode-optimization.html [last accessed 2015-12-22].

[89] Recommended upload encoding settings (Advanced) - YouTube Help. URL: https://support.google.com/youtube/answer/1722171?hl=en [last accessed 2015-12-22].

[90] What is HEVC? - High Efficiency Video Coding FAQ - Tom's Guide. URL: http://www.tomsguide.com/us/hevc-4k-explained,news-18206.html [last accessed 2016-01-07].

[91] Internet Connection Speed Recommendations. URL: https://help.netflix.com/en/node/306 [last accessed 2015-11-25].

[92] Samsung Galaxy S4 eye-tracking smartphone unveiled - BBC News. URL: http://www.bbc.com/news/technology-21791023 [last accessed 2016-01-07].

[93] MPEG-HEVC Support in GPAC | GPAC. URL: https://gpac.wp.mines-telecom.fr/2013/04/26/mpeg-hevc-support-in-gpac/{#}more-1038 [last accessed 2016-01-07].

[94] Browser Integration | GPAC. URL: https://gpac.wp.mines-telecom.fr/player/browser-integration/ [last accessed 2016-01-07].