FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automatic adaptation of views in 3D applications

João António Sequeira de Pinho

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Maria Teresa Andrade (PhD)

Co-supervisor: Tiago Costa (MsC)

July 22, 2013

Resumo

A auto-estereoscopia de multivista é uma tecnologia nova e desafiante que mudará o paradigma da representação 3D. Os sistemas 3D estão a aumentar a sua presença no mercado, afirmando-se como a tecnologia de vídeo do futuro próximo. Está no âmbito desta dissertação explorar e ajudar a aumentar o conhecimento nesta área, procurando novas formas de criar experiências personalizadas multivista, no âmbito do ImTV, um projeto de investigação atualmente desenvolvido no INESC-TEC, no âmbito do programa UTAustin-Portugal. Esta dissertação explora uma arquitetura cliente-servidor baseada em IP que muda a vista 3D automaticamente com base na posição do utilizador. De forma a enviar *streams* correspondentes a vistas de uma cena, foi escolhido o software de servidor - Sirannon - e em seguida adaptado para permitir a mudança de vista, entre outros requisitos. No lado do cliente foi desenvolvido um programa em C# que conjuga diferentes módulos. Estes módulos apresentam metadados relevantes, condições do sistema atuais e em conjunto com o FaceAPI concretizam a função de *headtracking*.

Abstract

Multiview auto-stereoscopy is a challenging new technology that will shift the paradigm of 3D representation. 3D systems are increasing their market share and solidifying themselves as the near future of video representation. It is the aim of this dissertation to explore and help to push forward the barrier of knowledge in this area by researching new ways of creating personalized multiview experiences, in the scope of ImTV, a project being developed in INESC-TEC, inserted into the UT-Austin collaboration. This work puts together an IP-client-server-architecture which enables the automatic change of the displayed 3D view, according to the head position of the user. In order to stream multiple views, Sirannon was chosen as the streaming server and its modules were adapted to allow stream switching and additional features. On the client side, a program written in C# joins different modules together. These modules display relevant metadata, current system performance and integrate with FaceAPI for the head-tracking capability.

Agradecimentos

Gostaria agradecer à minha orientadora, a Professora Doutora Maria Teresa Andrade, pelo desafio proporcionado em realizar esta dissertação em conjunto com o INESC-TEC, e, também, pelo acompanhamento num tema verdadeiramente interessante. Gostaria de exprimir agradecimentos ao meu coorientador, o Engenheiro Tiago Costa, pela simpatia, paciência e motivação com que me acompanhou e ajudou de perto na concretização deste projeto.

Agradeço à Faculdade de Engenharia da Universidade do Porto pela formação de qualidade que tive a oportunidade de receber ao longo do curso e ao INESC-TEC pelo estágio profissional que em muito me ajudará a fazer a transição para o mercado de trabalho.

Agradeço aos meus colegas de curso por me acompanharem e ajudarem ao longo do curso e em especial aos que colaboraram com ideias na elaboração deste documento.

Não podia deixar de agradecer aos meus amigos mais chegados pela amizade, força e pelos momentos de descontração nesta última fase e por último, mas certamente não de forma menos importante, aos meus pais e irmãs por me terem ajudado a ser quem sou hoje.

João de Pinho

"Actually, 3D is really the most normal thing because it's how those of us with two eyes usually see the world. TVs are the unusual things in 2D!"

Shigeru Miyamoto

Contents

Symbols and Abbreviations xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Objectives and Main Results . . . 3
- 1.3 Document Structure . . . 4
- 1.4 Website . . . 4

2 Overview of the developed solution 5
- 2.1 The Project . . . 5
- 2.2 Work methodology . . . 6
  - 2.2.1 Expected Results . . . 6
  - 2.2.2 Approach . . . 6
  - 2.2.3 Expected Challenges . . . 6
  - 2.2.4 Work Plan . . . 6
- 2.3 Functional Specification . . . 7
  - 2.3.1 Content Consumer (CC) Use Cases . . . 7
  - 2.3.2 Content Provider (CP) Use Cases . . . 8
- 2.4 Tool Specification . . . 8
  - 2.4.1 Headtracking . . . 8
  - 2.4.2 Media Server . . . 10
  - 2.4.3 Traffic analysis . . . 11
  - 2.4.4 Player . . . 12
- 2.5 System Overview of the proposed solution . . . 13

3 State of the Art 15
- 3.1 Human Visual System . . . 15
  - 3.1.1 Visual Confort and related issues . . . 16
- 3.2 3D Viewing Technologies . . . 17
  - 3.2.1 Stereoscopic . . . 17
  - 3.2.2 Autostereoscopic . . . 18
- 3.3 3D Coding Formats . . . 20
- 3.4 Video Compression . . . 22
  - 3.4.1 MPEG-C part 3 . . . 24
  - 3.4.2 H.264/AVC . . . 24
  - 3.4.3 H.264/MVC . . . 25
  - 3.4.4 H.265/HEVC . . . 26
- 3.5 Audio . . . 27

3.6 Network . . . 27
3.6.1 Client-Server networks . . . 27
3.6.2 P2P networks . . . 27
3.6.3 MPEG-TS . . . 28
3.6.4 Content Adaptation . . . 28
3.7 Conclusion . . . 28

4 Media Server **29**
4.1 The Sirannon Project . . . 29
4.2 Components and Modules . . . 29
4.2.1 Two view-pairs configuration for Multiview . . . 31
4.2.2 Three view-pairs configuration for Multiview . . . 31
4.2.3 View Change . . . 32
4.3 Media Files . . . 32

5 Client Interface **33**
5.1 Features . . . 33
5.2 Main Modules . . . 34
5.2.1 Headtracking . . . 34
5.2.2 Metadata . . . 35
5.2.3 Techdata . . . 36
5.2.4 3D Player . . . 36
5.2.5 Manual Modes . . . 38
5.3 Server Communication . . . 38
5.3.1 Headtracking communication . . . 38
5.3.2 Metadata communication . . . 39
5.3.3 Techdata communication . . . 39

6 Tests and Results **41**
6.1 Subjective tests . . . 41
6.2 Tests with the Iperf . . . 41

7 Conclusions **43**
7.1 Difficulties . . . 43
7.2 Future Work . . . 44
7.3 Final Remarks . . . 44

References **45**

List of Figures

1.1 3D TV share of global LCD TV panel shipments from the 2nd quarter of 2011 to the 4th quarter of 2012 in [4] (Source: Display Search, 2013 [5]) 2

2.1 Gantt Chart 6
2.2 System's Use Cases 7
2.3 System overview 8
2.4 System Architecture 13

3.1 Types of parallax 15
3.2 Accommodation and Convergence 16
3.3 Parallax Barrier Cross Section (Source: Jonathan Mather, 2012 [16]) 19
3.4 Lenticular layer placement 19
3.5 Multi-projector setup 20
3.6 Video plus depth data representation format consisting of regular 2D color video and accompanying 8-bit depth-images (Source: Müller, Merkle and Wiegand, 2007 [20]) 21
3.7 Simulcast versus MVV coding (Source: Müller, Merkle and Wiegand, 2007 [20]) 22
3.8 MVD versus LDV (Source: Bartczak, 2011 [21]) 22
3.9 Common sub-sampling modes (Source: Maria Teresa Andrade, 2011 [22]) 23
3.10 A sequence of Intra-coded, Predicted and Bi-predicted frames in a compressed video sequence. (Source: Petteri Aimonen, 2009 [23]) 23
3.11 V+D additional view generation (Source: Aljoscha Smolić, 2007 [25]) 24
3.12 Structure of an MVC bitstream including NAL units that are associated with a base view and NAL units that are associated with a non-base view. NAL Unit Types (NUT) indicators are used to distinguish different types of data that are carried in the bit stream. (Source: Vetro, Wiegand and Sullivan, 2011 [30]) 26

4.1 Example of connected modules in Sirannon 29
4.2 Example of a module's preference pane 30
4.3 Video Chain: from H.264 to transmission packets 30
4.4 Two view-pairs configuration in Sirannon 31
4.5 Three view-pairs configuration in Sirannon 32

5.1 Client Interface 33
5.2 Head position definition: translation, pitch, yaw, and roll (Source: How Stuff Works, 2007[40]) 34
5.3 Bino input layouts (Source: Bino Manual, 2011 [41]) 37
5.4 Bino output layouts (Source: Bino Manual, 2011 [41]) 37
5.5 Headtracking communication protocol 39

5.6 Metadata communication protocol 39
5.7 Techdata communication protocol 39

List of Tables

2.1 Headtracking Decision Matrix 10
2.2 Media Server Decision Matrix 11
2.3 Player Decision Matrix 13

5.1 Video metadata 35

6.1 Network load tests 42

Symbols and Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
API	Application Programming Interface
ASV	Asymmetrical Stereo Video
AVC	Advanced Video Coding
AVI	Audio Video Interleaved
bps	Bits per second
CSV	Conventional Stereo Video
DCT	Discrete Cosine Transform
DRM	Digital Rights Management
FCT	Fundação para a Ciência e Tecnologia
FVV	Free Viewpoint Video
GOP	Group of Pictures
HD	High-Definition
HDMI	High-Definition Multimedia Interface
HDTV	High-Definition Television
HEVC	High Efficiency Video Coding
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HVS	Human Visual System
INESC	Instituto de Engenharia de Sistemas e Computadores
IP	Internet Protocol
IPTV	Internet Protocol Television
LDV	Layered Depth Video
MKV	Matroska Multimedia Container file format
MOV	Quicktime Movie
MP4	MPEG-4 Part 14
MPEG	Moving Picture Experts Group
MPEG-PS	MPEG Program Stream
MPEG-TS	MPEG Transport Stream
MVC	MultiView Coding
MVD	Multiple Video plus Depth
MVV	MultiView Video
NAL	Network Abstraction Layer
P2P	Peer-to-peer
PES	Packetized Elementary Stream
PPS	Picture Parameter Sets
PSNR	Peak Signal-to-Noise Ratio

RAM	Random Access Memory
RGB	Red Green and Blue additive color model
RTMP	Real Time Messaging Protocol
RTMPT	Tunneled RTMP
RTP	Real time Transport Protocol
RTSP	Real Time Streaming Protocol
SEI	Supplemental Enhancement Information
SD	Standard Definition
SI unit	from the International System of Units (in french *Système international d'unités*)
SPS	Sequence Parameter Sets
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
TV	Television
UDP	User Datagram Protocol
V+D	Video plus Depth
VCEG	Video Coding Experts Group

Chapter 1

Introduction

This chapter contextualizes the dissertation presented in this document, providing relevant background information. It presents the challenges addressed, the motivation for overcoming them, the main goals to achieve and the methodologies to adopt in that process. It briefly indicates the expected outcomes of the developed work, highlighting strengths and weaknesses. Finally, it describes the structure of this document, listing the chapters and the main topics addressed in each one.

1.1 Motivation

In the truly fast evolving world of high-tech, television and broadcasting have suffered an incredible change in the past 60 years[1]. The arrival of the digital age has opened the path for ever increasing image quality and resolution. These changes were accompanied by a capture, coding and transmission revolution, enabling the display of scenes captured in HD quality or higher[2].

But high resolution may not always be sufficient. In fact the present reality of audiovisual communications embraces also the third dimension, enabling to provide a more realistic representation of the real world, which is naturally in three dimensions. Users' expectations have indeed raised beyond quality, to also include the sense of being inside the video scenes. This need for immersion opened the path for the return of the third dimension, experimented some decades ago without, however, much success. 3D video is quickly becoming the *status quo*, with movie theaters adopting stereoscopic equipment to display movies and equivalent home solutions made possible by the dissemination of high-speed internet connections[3] and the lowering price of stereoscopic displays, as shown in the 3D TV share of global LCD TV panel shipments from the 2nd quarter of 2011 to the 4th quarter of 2012 in Figure 1.1.

However, there are still some barriers to the fulfillment of 3D video as a true and natural representation of the real world and, thus to its adoption at large scale. In fact, current 3D technology offers an experience which is not as natural as desired. A first problem is due to the fact that the depth sensation is greatly dependent on the position of the viewer in relation to the screen.

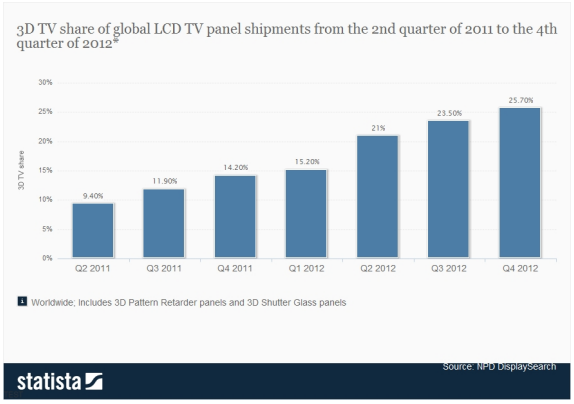

Figure 1.1: 3D TV share of global LCD TV panel shipments from the 2nd quarter of 2011 to the 4th quarter of 2012 in [4] (Source: Display Search, 2013 [5])

Secondly, viewers need to wear special glasses which, besides being cumbersome, also deteriorates the image quality. Two main domestic alternatives exist nowadays for visualizing stereo images. They both make use of special glasses that are able to provide a separate image to each eye. The brain then combines the two images into a single image with 3D characteristics. The Passive method requires the use of passive polarized glasses, where each lens has a different polarization filter (Linear or Circular Polarization Systems) or a color filter (Anaglyph Systems), thus delivering to each eye a slightly different image. The Active shutter technology requires the use of glasses with embed electronics to synchronize moving LCD shutters, placed in each lens, with the onscreen display rate. When comparing the two alternatives, the major disadvantage of the former is that the resolution of the 3D image is halved in relation to that of a 2D representation. In turn, the major disadvantage of the latter is that many viewers experience flickering and fatigue due to the opening and closing of the LCD shutters. Additionally, active glasses are considerably more expensive than the passive ones. As it can be seen, the current alternatives for providing a 3D experience to the user have major drawbacks, leading the scientific community to endorse efforts in finding better solutions. In fact, during the last decade, considerable efforts have been undertaken towards the development of a technology that could provide an immersive 3D viewing experience to overcome the existing limitations. The result was the development of the so-called auto-stereoscopy display technology which, through the use of Lenticular Lenses and Parallax Barrier, enables the viewer to enjoy a more natural, glass-free 3D experience.

Critics of the 3D technology claim that the only right way of showing 3D video is through holographic methods, adopting the same principles of holographic images (holograms) that exist for a long time now. In fact, holography seems to be a more faithful correspondence to the natural way our eyes work for visualizing an object, especially when compared to the currently used 3D technologies: unlike the latter, that require each eye see a distinct image, with holographs both eyes can converge to the same point in space, which is the natural way for a human being to visualize objects in the real world. Indeed, 3D objects in holographic video are formed when different light beams are projected out of the screen to converge to the point in space where the object

was supposed to be. And both our eyes also converge to that same point to have the perception of the 3D object. Although holographic video can indeed provide a truly immersive experience, this technology is still in its infancy, as great limitations still exist to be able to reproduce a wide spectrum of colors and fast movements. One of the biggest limitations is the need for projecting simultaneously a great number of different patterns to allow visualizing the object from different angles. The other one is the need for having very reduced pixel dimensions.

A possible alternative to offer the sense of immersion, enabling 3D viewing without glasses and from multiple angles (allowing the viewer to move in front of the screen) is to resort to auto-stereoscopic displays and to deliver to the display multiple views of the same scene, shot from different angles. Multiview Video Coding (MVC) is a 3D coding algorithm developed by MPEG that allows the building of 3D immersive environments. It is based on the video algorithm AVC/H.264 and it is in fact an extension to this standard. At the source, multiple views of the same scene are captured from different angles using an array of cameras. Then, the encoding algorithm compresses those views taking advantage of redundancy that exists between the multiple views. When all the views are sent to an auto-stereoscopic display it enables wide-field-of-view presentations. However, as all captured views must be sent to the receiver, the bandwidth requirements are extremely high in spite of the compression applied. Moreover, the display would need to have very high refresh rates to be able to present all the received views in each eye at a comfortable rate. These two limitations have fostered the development of Free Viewpoint Video (FVV) applications. FVV is based on H.264-MVC and it empowers the client to regenerate views of the scene that have not been transmitted. It thus enables it to transmit a limited number of views to the display, therefore achieving network savings. However, it significantly overloads the client, as it requires rather complex processing capability in order to identify the desired viewing angle and accordingly regenerate the corresponding missing view.

1.2 Objectives and Main Results

For a truly immersive experience, using additional 3D views can indeed improve the realism. To allow the viewer to freely select the desired view of the object, scenes are shot with a multiview setup. An automatic selection and adaptation of views is highly desired, as it leaves the viewer with a greater sense of immersion.

Unfortunately, access to technology is not always affordable, as referred previously. Even though stereoscopic displays' price is converging with 2D HD ones [6], auto-stereoscopy is still far from what consumers are willing to pay. Multiview implementations have the added issue of bandwidth consumption (when transmitting all the views) and in order to display them the refresh rates need to be very high (300/400Hz), which does not only greatly increases the implementation difficulty but also the final cost.

The work that was proposed to be developed in this dissertation reflects the desire to seek feasible and cost-effective alternatives to provide personalized immersive experiences to the user. The idea was to investigate the possibility of using MVC to produce a multiview representation

of a scene and to automatically select a given view to be sent to the client according to the user's visual focus of attention. The envisaged solution should be able to automatically recognize the area of interest of the user in the scene being displayed, sending that information to the server that would consequently switch between views to better satisfy the user's interests. Bypassing the bandwidth issue, which can be done by choosing the required view-pair or a mix of full-views and sub-sampled views according to the current need, is one of the main motivations of this dissertation.

3D systems are increasing their market share and solidifying themselves as the near future of video representation. Multiview auto-stereoscopy is a challenging new technology that will shift the paradigm of 3D technology. It is the aim of this dissertation to explore and help pushing forward the barrier of knowledge in this area by exploring new ways of creating personalized multiview experiences.

To attain this objective, a client-server multiview system is being developed (within the context of the ongoing research project ImTV described in section 2.1), having as basic components a Media Server with advanced streaming capabilities and a head-tracking client with a web-camera with an embed processing capability. This system enables the transmission of views across an IP network from a multiview server to a terminal with an autostereoscopic display that changes the view displayed according to the inferred user's visual attention as indicated by the tracking mechanism.

1.3 Document Structure

This document is organized in seven chapters, each of them starting with an introduction explaining its organization and theme. Chapter 2 provides an overview of the personalized multiview system, which was partly developed within the context of this dissertation and conceived as part of the objectives of the ImTV project. This chapter also describes the work methodology adopted during the progress of this dissertation, the work plan followed by the tools used to fulfill the proposed objectives. A thorough state of the art is presented in Chapter 3, focusing on the most important aspects of 3D video. The fourth chapter gives an extended view of the implemented Media Server and the nature of its original project. The Client Interface is presented on Chapter 5, followed by Tests and Results on Chapter 6. In the last chapter, overall difficulties, conclusions and future work suggestions are given.

1.4 Website

In order to better organize and to provide more detail about the dissertation work, a website is available at `http://paginas.fe.up.pt/~ee07241`. There, a project description and planning with detailed activity progress can be found. Used tools, methodology, documents and references are available, as well as relevant contact information about the team behind the project.

Chapter 2

Overview of the developed solution

This chapter provides an overview of the work developed during the course of this dissertation, contextualizing it within the scope of the on-going research project ImTV, being currently developed at INESC Porto. It starts by briefly describing the objectives and scope of ImTV. It then presents the work methodology, indicating the planning drawn for the development of the proposed work. It finally provides a description of the functional specification of the developed system, which can be seen as a component of the complete prototype being implemented in the ImTV project.

2.1 The Project

The dissertation work is within the context of a research project currently being conducted by the Telecommunications and Multimedia Unit of INESC Porto, the ImTV project.

ImTV - On-demand Immersive TV - is a collaborative research project, funded by the FCT under the Portugal-Austin Program (http://utaustinportugal.org). ImTV's main focus is exploring new trends in Media production and consumption, focusing on on-demand immersive television. This project takes into consideration the needs of the TV industry and the end-users' interests. Said consideration is present in its goals: understand future trends, shorten the production-to-distribution road and engage users.

In INESC-TEC, an Associate Laboratory coordinated by INESC Porto, the current outlines of ImTV involve developing a client-server multiview system. This system comprises of a Media Server with streaming capabilities and a headtracking client. The objective is to enable the transmission of different 3D views of a scene in an IP network to a terminal with an auto-stereoscopic display. The view change is done according to the user tracking result, registered via a web-camera.

2.2 Work methodology

2.2.1 Expected Results

It is the goal of this dissertation to create solutions for an effortless and enjoyable multiview experience. As the user changes their head positioning, the system should alter the displayed view to the corresponding one with the minimum latency possible and in a way that appears natural, for maximum immersion. In order to achieve such goals, the system should choose where in the bit-stream the view transition is appropriate (minimizing the impact on the quality perceived by the user), what triggers a view change, how the views will be in terms of compression and quality, what are the best transmission techniques and which is the most relevant metadata for transmission.

2.2.2 Approach

The chosen approach to achieve the expected results consists of familiarizing with 3D video through a state-of-the-art digest, finding the best software and tools for both the client and the server via a market survey and testing the developed solution by running quality and robustness tests.

2.2.3 Expected Challenges

At first glance, the most challenging aspects of this dissertation are finding the best transition spots in the bitstream and quickly giving the user the highest resolution possible immediately after the view transition. From an implementation point of view, the programming languages and their integration into the system were what appeared to be the most challenging.

2.2.4 Work Plan

The Dissertation Work followed the work plan on the Gantt Chart in Figure 2.1.

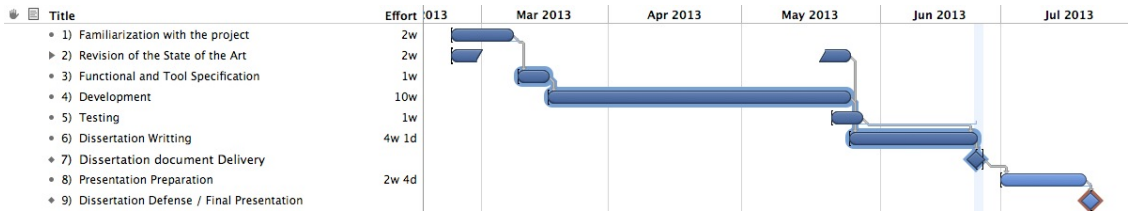

Figure 2.1: Gantt Chart

The initial work stage consisted of researching the state-of-the-art on the main subjects at hand, analyze the functional specification of the system describing its expected behaviour and deciding which tools should be used. The second and most extensive stage was development, which was divided into an initial familiarization with the existing system at INESC, followed by modifying the server and creating a client interface. Testing the developed solution was the final stage.

2.3 Functional Specification

In this section, the results of a functional analysis are shown via the use cases the final system should achieve, shown in Figure 2.2. Some of these Use Cases fall beyond the scope of this dissertation mainly due to time constraints (signaled with a *).

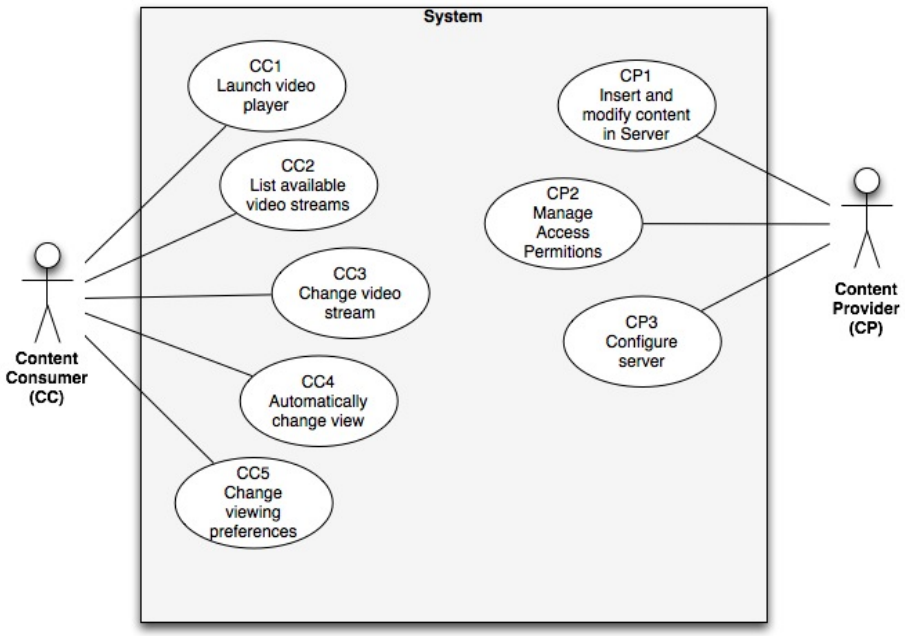

Figure 2.2: System's Use Cases

2.3.1 Content Consumer (CC) Use Cases

These are the Content Consumer Use Cases, which focus mainly on features on the Client side, depending nonetheless on the Server.

CC1 - Launch Video Player - The CC opens the client interface and launches the video player, beginning to visualize the video stream.

CC2 - List available video Streams* - The CC requests the Server a list of available stream views.

CC3 - Change video stream - According to the results of CC2, the CC selects the desired video stream view.

CC4 - Automatically change view - Based on the results of headtracking, the main 3D view changes.

CC5 - Change viewing preferences - The CC can change the current viewing preferences: video resolution/quality.

2.3.2 Content Provider (CP) Use Cases

These use cases focus on the Content Provider, which will interact directly with the Media Server.

CP1 - Insert and modify content in Server - The CP can insert new video streams, replace them with others or delete them on the server.

CP2 - Manage Access Permissions* - Restrict video stream access to specific or groups of CCs (DRM).

CP3 - Configure server - Manage technical parameters of transmission: set maximum resolutions/quality, set maximum latency*, minimum network channel quality.

2.4 Tool Specification

In this section, a specification of the tools used in the system is made, based on the results of the functional analysis. The tools choosing method consisted in making a market survey for the main parts of the system. Tools were rated in terms of relevant parameters and those parameters were ordered regarding its given importance to the system. The maximum value of the weighed sum of the parameter rating and its weight determined the tool choice. Figure 2.3 shows a simple overview of the system.

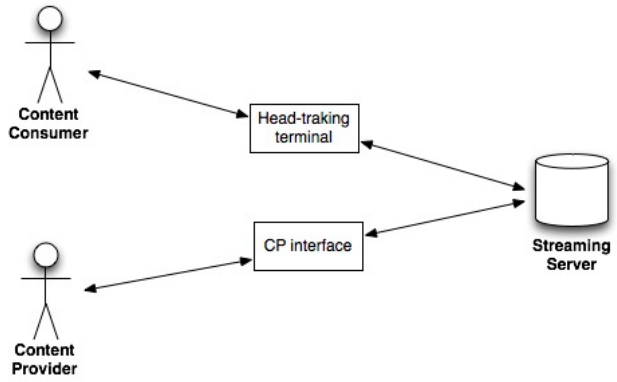

Figure 2.3: System overview

2.4.1 Headtracking

On the Content Consumer side, a headtracking technology must transmit the CC's current position.

2.4.1.1 Requirements

The following requirements are weighted (W) from 0 to 5. The scoring to each requirement for each solution is from 0 to 5.

- Open Source or an API is provided, so modifications are possible. W: 5
  Scoring: 0 - no API provided, proprietary software; 3 - a minimal API is given or partial code is open source; 5 - all code is open source or API is extensive and allows full interaction with code

- Non-intrusive, for maximum comfort. W: 4
  Scoring: 0 - discomfort is necessary for the recognition to work; 1 - some additional gear is used, which can cause some discomfort; 3 - initial calibration necessary using physical markers of some sort; 5 - completely non-intrusive

- Multi-platform, provides liberty when choosing operating system. W: 4
  Scoring: 0 - only one platform is supported; 3 - two popular platforms are supported; 5 - the three most popular platforms are supported (Windows™, Linux™and MacOS®)

- Degrees of freedom detected from the user, the number of variations recognizable. W: 2
  Scoring: 0 - only yaw is detected; 3 - yaw and another degree is detected; 5 - yaw, pitch and roll are detected

- Camera compatibility. W: 3
  Scoring: 0 - only one specific make and model is supported; 3 - only one brand is supported; 5 - most common types of cameras are supported

- Ease of use - out of the box. W: 4
  Scoring: 0 - compilation and/or major code modifications are required pre-installation; 3 - some minor tweaks are necessary before or during installation; 5 - fully functional program with only minor configuration following the installation

- Overall performance and robustness. W: 3
  Scoring: 0 - constant program crashes and bugs; 3 - some sporadic crashes and performance issues; 4 - minor bugs and crashes, overall good performance; 5 - overall excellent performance, typically not prone to crashes

2.4.1.2 Found Solutions

- OpenCV - `http://opencv.willowgarage.com/wiki`
- Cachya - `http://www.cachya.com/esight/overview.php`
- FaceAPI - `http://www.seeingmachines.com/product/faceapi`

2.4.1.3 Decision Matrix

Evaluating these solutions showed that OpenCV and FaceAPI were the most indicated solutions. The ease of use and the better performance of **FaceAPI** determined its choosing as the headtracking solution.

Headtracking Decision Matrix	W	OpenCV	Cachya	FaceAPI
Open Source or API	5	5	0	5
Non-intrusive	4	5	1	5
Multi-platform	4	5	0	0
Degrees of freedom	2	5	5	5
Camera compatibility	3	5	5	5
Ease of use	4	0	5	5
Overall performance and robustness	3	4	4	5
Score		102	61	**105**

Table 2.1: Headtracking Decision Matrix

2.4.2 Media Server

On the Server Side, the Media Server must stream media and allow view and stream change operations.

2.4.2.1 Requirements

The following requirements are weighted (W) from 0 to 5. The scoring to each requirement for each solution is from 0 to 5.

- Open Source: so additional features which require code modification can be made. W: 5
  Scoring: 0 - proprietary software; 3 - partial code is open source; 5 - all code is open source

- Communication Protocol Support (UDP, RTP, ...). W: 4
  Scoring: 0 - no streaming standard protocols supported; 3 - proprietary or platform-specific protocols; 4 - common protocols supported, some modification may be necessary for full functionality; 5 - all major streaming protocols supported with no required modifications

- Video Format Support (MP4, H.264). W: 4
  Scoring: 0 - no support for MP4/H.264; 3 - support for either MP4 or H.264; 5 - full support for MP4 and H.264

- Stereoscopy support. W: 5
  Scoring: 0 - only one stream is transmittable; 5 - multiple concurring streams can be sent

- Multi-platform. W: 3
  Scoring: 0 - only one platform is supported; 3 - two popular platforms are supported; 5 - the three most popular platforms are supported Windows™, Linux™and MacOS®)

- Documentation. W: 4
  Scoring: 0 - minimum documentation is provided; 3 - some documentation is provided along with basic examples or tutorials; 5 - comprehensive documentation, examples and tutorials provided

2.4.2.2 Found Solutions

- Live555 - http://www.live555.com
- Sirannon - http://sirannon.atlantis.ugent.be
- IIS Smooth Streaming - http://www.iis.net/downloads/microsoft/smooth-streaming
- Red5 - http://www.red5.org/

2.4.2.3 Decision Matrix

Table 2.2 shows the decision matrix for the Media Server.

Media Server Decision Matrix	W	Live555	Sirannon	IIS Smooth Streaming	Red5
Open Source	5	5	5	3	5
Communication Protocols	4	5	5	3	5
Video Formats	4	5	5	5	3
Stereoscopic Support	5	5	5	5	5
Multi-platform	3	5	3	0	5
Documentation	4	3	5	3	3
Score		117	**119**	84	109

Table 2.2: Media Server Decision Matrix

Of the studied solutions, the final decision was between Live555 and Sirannon. Based on the documentation provided (with tutorials and examples), **Sirannon** was chosen.

2.4.3 Traffic analysis

The following tools are needed to help develop, test and analyze the communication between Client and Server.

2.4.3.1 Wireshark

As a way of analyzing and troubleshooting the exchanged network packets, Wireshark, the most used tool for protocol analysis and network research, was the obvious choice. Wireshark is a free, open-source, cross-platform and basically a graphical, more complete and refined version of *tcpdump*, a comand-line packet analyzer. Wireshark not only captures and displays the packets but has powerful sorting and packet filtering [7].

2.4.3.2 Iperf

In order to analyze the effects of network load on the system performance, the solution had not only to interpret network congestion but also create it. Iperf was the ideal solution, being a C++ tool that creates TCP and UDP data streams and measures the available bandwidth, jitter (variation

of the delay) and the lost packets percentage. Iperf can work in unidirectional and bidirectional mode and it is installed on both ends of the system [8].

2.4.4 Player

On the Client Side, the player must be able to receive and display the video stream in 3D.

2.4.4.1 Requirements

The following requirements are weighted (W) from 0 to 5. The scoring to each requirement for each solution is from 0 to 5.

- Stereoscopy support. W: 5
  Scoring: 0 - only 2D is displayable; 1 - Side-by-Side video support; 5 - full stereoscopic support
- Stereoscopy transformation. W: 4
  Scoring: 0 - no transformation options are given, video is displayed as received; 5 - possibility of selecting different types of 3D modes
- RTP / UDP support. W: 5
  Scoring: 0 - no support for RTP/UDP streaming; 4 - indication of possible full support (not validated in tests yet); 5 - verified full support for RTP/UDP
- Multi-platform. W: 4
  Scoring: 0 - only one platform is supported; 3 - two popular platforms are supported; 5 - the three most popular platforms are supported (Windows™, Linux™and MacOS®)
- Open Source. W: 4
  Scoring: 0 - proprietary software; 3 - partial code is open source; 5 - all code is open source
- Hardware Independence. W: 3
  Scoring: 0 - fully dependent on specific hardware (graphic card, display, glasses or other); 5 - fully hardware independent

2.4.4.2 Found Solutions

- VLC - `http://www.videolan.org/vlc/index.html`
- NVidia 3D Vision Video Player - `http://www.nvidia.com/object/3d-vision-video-player-1.7.2-driver.html`
- Stereoscopic Player by 3dtv.at - `http://www.3dtv.at/Products/Player/Features_en.aspx`
- Bino - `http://bino3d.org`

2.4.4.3 Decision Matrix

Table 2.3 shows the decision matrix for the media player.

Player Decision Matrix	W	VLC	NVidia 3D Vision	Stereoscopic Player	Bino
Stereoscopy Support	5	1	5	5	5
Stereoscopy transformation	4	0	5	5	5
RTP / UDP support	5	5	0	0	4
Multi-platform	4	5	0	0	5
Open Source	4	5	0	0	5
Hardware Independence	3	5	0	5	5
Score		85	45	60	**120**

Table 2.3: Player Decision Matrix

Although VLC has proven superior even with malformed or mistagged UDP packets, **Bino** offers the stereoscopic support that VLC is lacking at the moment.

2.5 System Overview of the proposed solution

As mentioned before, the system will follow a Client-Server architecture. Based on the decisions made on this chapter, the diagram in Figure 2.4 shows the proposed system architecture. Each side (Client and Server) uses RTP for the video transmission/reception and UDP-based modules for the other data exchanged. The UDP modules are separated into Headtracking, Metadata and Techdata. Further information on these specific modules is given in section 5.2.

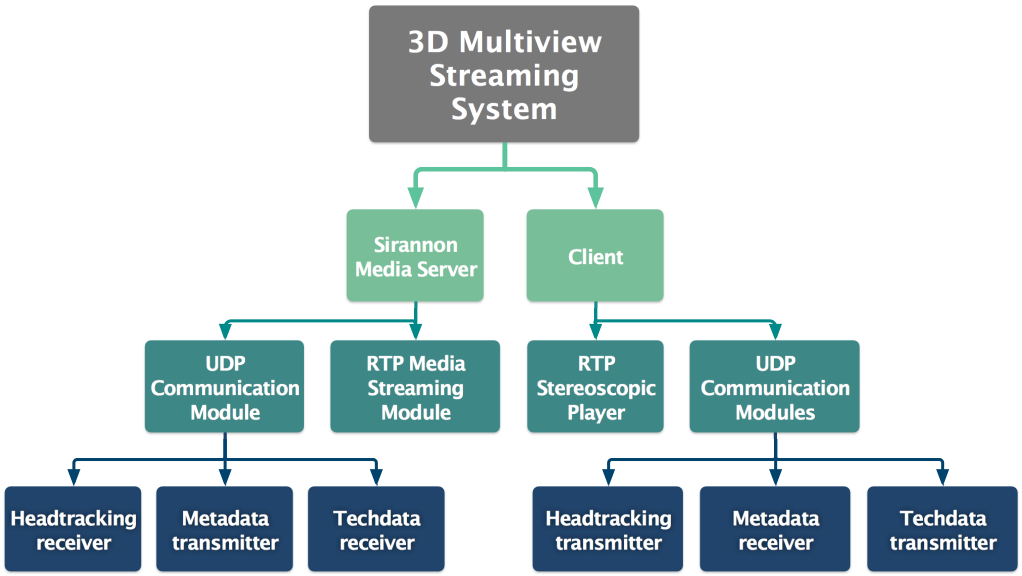

Figure 2.4: System Architecture

Chapter 3

State of the Art

In this chapter, a comprehensive view on all 3D matters is given, starting with the Human Visual System - understanding how we interpret 3D images and how that can become uncomfortable - followed by an overview of 3D viewing technologies. After going through the ways of coding 3D video, a review of coding standards is done. Audio is mentioned briefly, followed by transport and content adaptation.

3.1 Human Visual System

Human perception of depth, or the ability to distinguish the relative position and size of objects, is possible due to the horizontal separation of the two eyes (average of 65mm) [9]. This separation leads to the acquisition of two slightly different images which are then processed by our brain's visual system, creating a sense of depth.

This mechanism is taken advantage of in 3D systems, with the positioning information of objects being coded based on that perception. When recording 3D images with a double camera system there are some aspects regarding the HVS that should be observed: interocular distance, types of parallax (negative, zero or positive) and the geometry setup (toed-in or parallel).

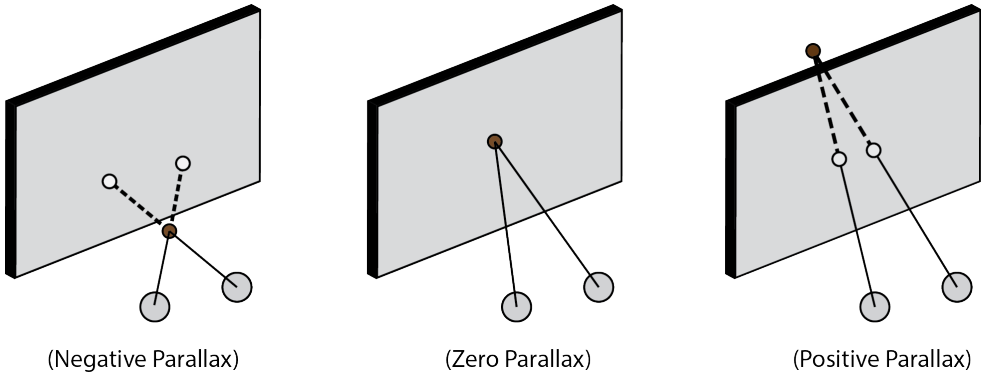

Figure 3.1: Types of parallax

In figure 3.1, three different types of parallax can be observed: Negative parallax, when the object is perceived to be between the screen and the viewer (*viewer space*), zero parallax, when both eyes perceive an object at the depth level of the screen (*screen space*) and positive Parallax, when the object is seen beyond the *screen space*.

3.1.1 Visual Confort and related issues

"Now that the demand for 3D-TV services is becoming stronger, the concerns related to the safety and health of viewing stereoscopic images have taken on a more prominent role. Some researchers have even raised the question of whether intensive watching of stereoscopic imaging could bring harm to viewers, especially to children whose visual system is still under development and, thus, probably more susceptible to external influences."

in Stereoscopic 3D-TV: Visual Comfort by Tam, W.J et al. [10]

Visual Comfort should always be a priority when working on 3D systems. The more uncomfortable experience created by the visual effects, the less viewers want to use 3D systems. One of the reasons why in the first *Golden Age of 3D* (1950s) viewers felt visual discomfort was due to the disrespect for the relation between accommodation and convergence [10]. Movie makers, trying to show off this spectacular new technology, would often create sudden object transitions between viewing spaces (viewer, screen and beyond screen), creating visual confusion and discomfort.

3.1.1.1 Accommodation / Convergence

When visualizing 2D content, our eyes keep focused at screen distance (accommodation) and both eyes converge to the same point, also at screen distance (convergence). But when 3D is introduced, the accommodation stays at screen level but the convergence is at the level of the object at focus, as depicted in Figure 3.2.

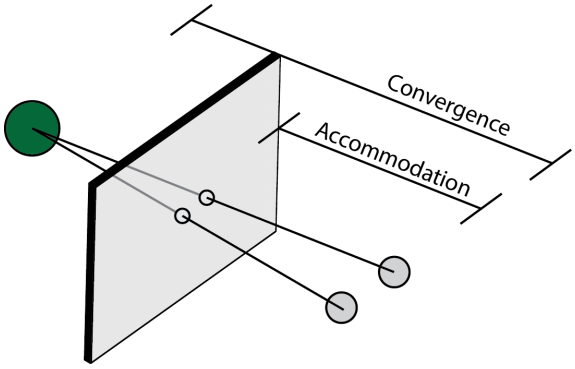

Figure 3.2: Accommodation and Convergence

If the disparity between accommodation and convergence is too great or too sudden, the eyes might try to adjust and move the accommodation towards the object level, which in turn will blur the perceived object (as it is in reality at screen level). As the object gets blurry, the eyes will then

try to accommodate towards screen level. These two diverging demands are known to be the cause of eye strain and discomfort. In order to avoid these issues, the perceived depth of objects should fall in the *comfort zone* [11] - between 0.2 and 0.3 diopters.

3.1.1.2 Parallax Distribution

One interesting finding about visual comfort came from *"Parallax distribution and visual comfort on Stereoscopic HDTV"* by Y. Nojiri et al. [12] where it is stated that visual comfort is strongly linked to how much and where in the screen parallax was used.

Interestingly, it is suggested that the most comfortable parallax setup for stereoscopic images is when it is negative at the bottom of the screen and positive at the top. One expected finding was that the less amount of parallax and motion an images has, the more comfortable it is.

3.1.1.3 Binocular Mismatches and Depth Inconsistencies

Alignment errors between the two sides that compose a stereoscopic image and the erroneous mapping of depth can be the cause of visual discomfort. Alignment errors can happen due to different errors in the chain: camera misalignment, lenses differences or distortion caused by the camera configuration during the capture process, luminance, color or sharpness editing errors, or even at the end of the process - miscalibrated display equipment.

Depth inconsistencies can be the result of errors in the depth map, where this information is stored. Errors in the depth map usually originate in the editing process (compression for instance) or in the transmission process, producing artifacts.

3.2 3D Viewing Technologies

There are currently multiple ways of displaying 3D video, each with their own advantages and drawbacks.

3.2.1 Stereoscopic

Stereoscopic systems date back to the 19th century when anaglyph 3D representations were invented, although it was not until the beginning of the 20th century that it gained visibility with the popularization of motion pictures.

Anaglyph systems separate the two views by placing the view data in a specific color channel (usually red and blue) which is then decoded by the use of color filters in each eye. Because of the system's limitation in color reproduction and induced visual discomfort from channel crosstalk, it was later replaced by polarized systems in the 1950s. Nevertheless, work in wavelength-division multiplexing continued to be developed and multi-band wavelength division multiplexing systems, like those by Infitec GmbH, were created. In these systems, the color spectrum is split into complementary sets, enabling good color representation. [13]

Due to the color representation issues in the anaglyph systems used, polarizing multiplexing techniques replaced those systems in movie theaters in the 1950s. Light polarization can reliably represent color but requires a non-depolarizing screen, the projection system must use polarizing filters (so do the viewers) and viewers must be on the viewing angle (usually narrow). Linear polarization (views separated using horizontal and vertical polarization) has some intrinsic problems, mainly the issues regarding the viewer's head position - if the head is tilted the polarization could fail. But what led to the decline of the usage of this technology in the first *Golden Age of 3D* were the inherent synchronization problems, as the two projectors had not only to be aligned but completely synchronous, otherwise it would not work. Other problems were related to the need of greater projection lightening - 30% more - due to the light that is blocked by the polarizing filters [14].

Stereoscopic systems are the *Status Quo* of 3D systems currently available in the market. The two views composing the stereoscopic image are either displayed at the same time with orthogonal light polarization (circular or linear) - typically in movie theaters - or in alternate frames, time-multiplexing using at least doubled frame-rate - usually at home. In order to separate the views for each eye, passive glasses (with different polarizing filters) or active ones (active-shutter glasses, which alternates frames between the two eyes via a shutter) must be used. Active glasses can be the answer to the image alignment and projection issues, but they also require synchronization and a refresh rate of usually 120Hz or double the original original refresh rate[13].

3.2.2 Autostereoscopic

Autostereoscopic systems not only provide a glassless experience but avoid certain issues linked with stereoscopic displays, such as projector alignment errors. Multiple solutions for separating views were created and are able to send information of each view to the corresponding eye.

Autostereoscopy solutions project the views to specified locations, supposing the viewer has a fixed location or locations. Two of these types of solutions use Parallax Barrier or Lenticular Displays to create multiple viewing zones. Both Parallax Barrier and Lenticular based systems can produce the effects of inverse stereoscopy (pseudoscopy) and cause visual discomfort. According to Dodgson, N.A. [15] there is a 50% chance of occurrence. One type of autostereoscopy that can resolve this issue is Multi-projector.

3.2.2.1 Parallax Barrier

Parallax Barrier systems are composed of a flat screen with a layer of opaque stripes that allow each eye to only see the corresponding set of pixels. This layer follows a set of parameters specific to that display and estimated viewer distance (pixel pitch, slit width and pitch, barrier distance from the screen surface) - explained in Figure 3.3.

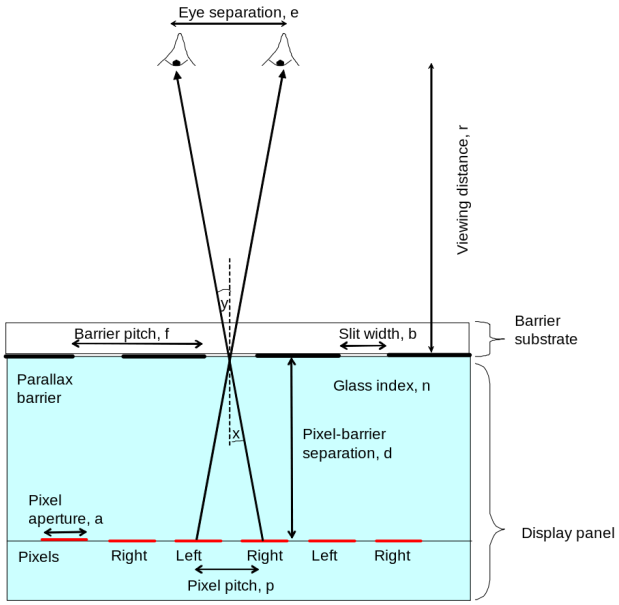

Figure 3.3: Parallax Barrier Cross Section (Source: Jonathan Mather, 2012 [16])

3.2.2.2 Lenticular Displays

Lenticular displays project their views using cylindrical lenslets in front of the display's pixels. Each lenticule has electrically-activated liquid crystals, which allow choosing between 2D and 3D mode [9]. Early implementations aligned the lenticular layer vertically to the display, which created a loss of resolution. In order to bypass this, the lenticular layer is now placed at a slight angle (slanted lenses), distributing the resolution loss, as shown in Figure 3.4.

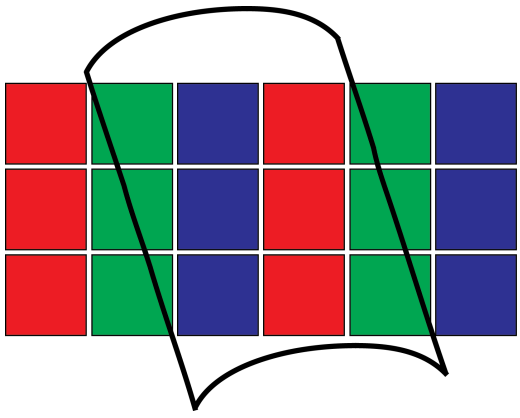

Figure 3.4: Lenticular layer placement

3.2.2.3 Multi-projector

Multi-projector systems use an array of carefully aligned projectors in which each projects a view into a lenticular screen which can be double sided or reflective, if the system has rear or front-projection respectively [17]. Aligning the array of projectors is not an easy task and because it defines the viewing zone it is of the utmost importance. Usually an independent camera is used to aid the alignment. The number of views depends on the number of projectors. According to Dodgson, N.A. [15] there are experimental systems with over 100 views.

One main advantage of multi-projector systems is scalability, which unfortunately comes with one of the main downfalls of this sort of system - its cost.

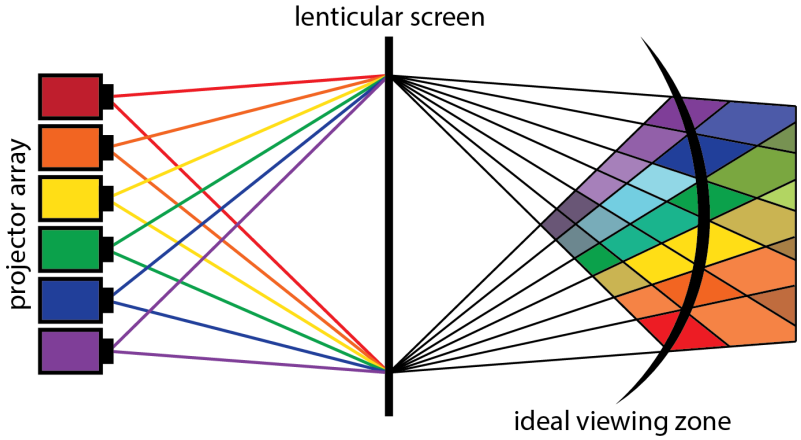

Figure 3.5: Multi-projector setup

3.2.2.4 Volumetric

Volumetric systems give the illusion of a three-dimensional representation through the projection of light to a moving object (light is projected into a revolving panel) or a static object (light is projected into multiple planes). Besides supporting multiple viewers and a 360º viewing angle, these systems don't suffer from the issues related to accommodation versus convergence. On the down side, no background can be represented and, because the viewing angle is so wide, the capture and coding of images presents additional challenges. These limitations make such systems not ideal for the consumer market, at least for now. One interesting implementation is the one developed by the Graphics Lab [18] at the University of Southern California.

3.3 3D Coding Formats

Displaying stereoscopic images involves sending at least one more perspective than 2D. When considering multiview systems, the importance of effective coding greatly increases. As the amount of transmitted and processed data grows with the number of views, it can quickly reach very high levels.

Conventional Stereo Video (CSV)

The most simple and straightforward approach to stereoscopy is Conventional Stereo Video. The two perspectives are individually encoded and transmitted. Despite its simplicity and low computational requirements, no redundancy is eliminated and both perspectives are sent in full-resolution, making this a very resource demanding way of coding 3D images transmission and storage wise.

Asymmetrical Stereo Video (ASV)

Asymmetrical Stereo Video (ASV) is based on the HVS' suppression theory which states that if one view has full quality, the other can have its resolution or bit-rate reduced without compromising the perceived quality. According to Saygili, G et al. [19], creating the asymmetry using Peak Signal-to-Noise Ratio (PSNR) rather than reducing the spatial resolution provides a better control of the output's PSNR and a better perceived quality.

Video plus Depth (V+D)

Video plus Depth uses a 2D signal and the corresponding depth, be it estimated or recorded by cameras. The 2D signal is of full resolution and the depth layer is typically an 8-bit scaled quantized representation (Figure 3.6). Because the depth layer has a lot less information than the 2D view and the redundancy that exists in Conventional Stereo Video is eliminated, coding efficiency is achieved.

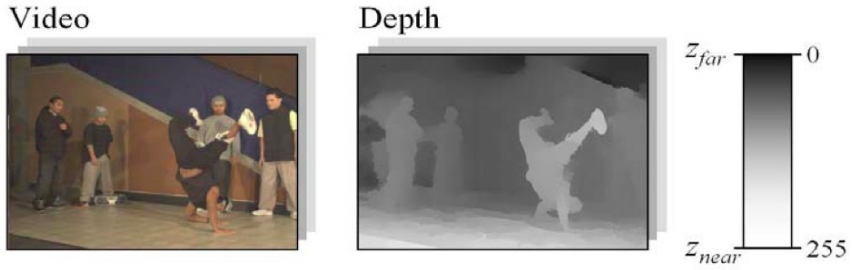

Figure 3.6: Video plus depth data representation format consisting of regular 2D color video and accompanying 8-bit depth-images (Source: Müller, Merkle and Wiegand, 2007 [20])

Multiple Video plus Depth (MVD)

Concatenating 3D views coded in V+D (simulcast) might be the most simple approach, but not the most efficient. Redundancy between views can be explored if views are coded together as shown in Figure 3.7.

Multiview video (MVV) coding not only takes advantage of the temporal redundancy but also deals with the inter-view redundancy. The views are separated in the decoding process. Compression efficiency is highly correlated to the type of scene but overall MVV coding performs more efficiently than simulcast coding [20]. As occurs in single view stereoscopic

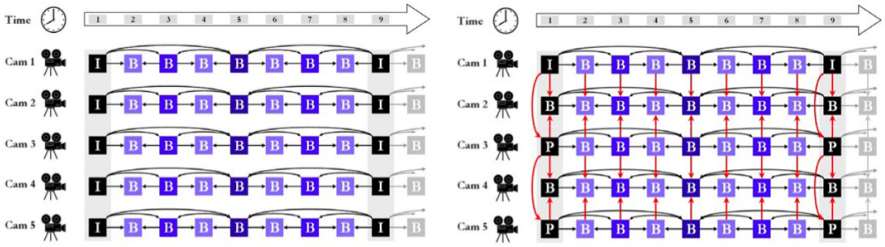

Figure 3.7: Simulcast versus MVV coding (Source: Müller, Merkle and Wiegand, 2007 [20])

images, MVV coding can be improved by adding the same strategy of V+D, hence creating Multiple Video plus Depth (MVD).

Layered Depth Video (LDV)

Exploring temporal and inter-view redundancy can increase efficiency, but there is still information that the multiple viewpoint camera system creates that is not explored - such as foreground and occluded elements. It is the coding of occluded objects that provides the biggest advantage of using Layered Depth Video (LDV). This coding format stores the foreground color and depth, along with the information of the occluded elements, which are analyzed for redundancies based on a set reference view - center view for example in Figure 3.8. Because of the simple way LDV handles these elements, the coding efficiency is high [21].

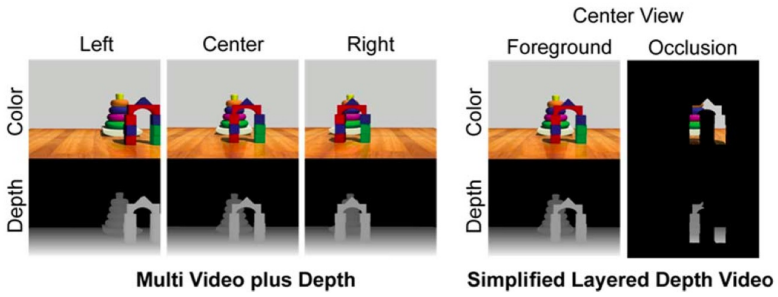

Figure 3.8: MVD versus LDV (Source: Bartczak, 2011 [21])

3.4 Video Compression

The efficiency of the coding formats mentioned in the previous section (3.3) would be hindered if raw video was used. Using video compression becomes pivotal when considering video transmission. For instance, a full HD video (1080p resolution, or 1920x1080 pixels) captured at 50 frames per second and using a 24bit color depth (Bayer mask) would generate a constant bitrate of 2,49 Gbps (SI unit).

Video compression can be divided into lossless - the product of this compression allows a full reconstruction of the original file, works by reducing statistical redundancy - and lossy - information is discarded, trying to balance quality and data used. Lossy video compression works by taking advantage of the HVS in three main vectors of redundancy elimination: spectral (color spaces), temporal (motion estimation, compensation) and spacial (block DCT).

Spectral redundancy can be reduced by color sub-sampling. The YC_BC_R, a way of encoding the RGB space is used (Y being the luminance, C_B and C_R are the chrominance components in blue-difference and red-difference correspondingly). A Y:C_B:C_R, notation is used, and the most common modes of color sub-sampling are shown in Figure 3.9.

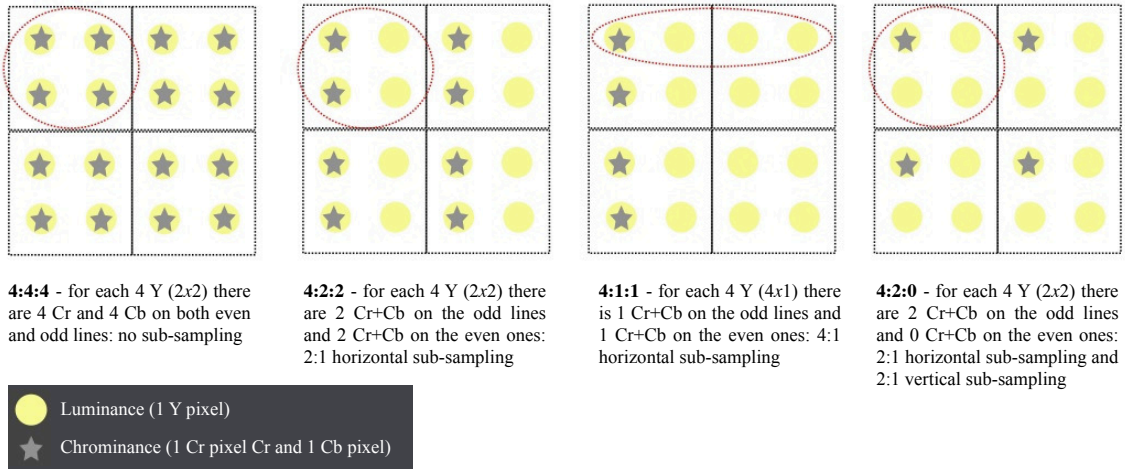

Figure 3.9: Common sub-sampling modes (Source: Maria Teresa Andrade, 2011 [22])

Reducing temporal redundancy is done by estimating the vector of movement in pixel blocks between adjacent video frames. This information is then used to perform motion compensation, in which frames are synthesized from reference frames called Intra (I-frames, full information). Predictive frames - P-frames - only contain the coded difference from the previous frame and Bi-predictive-frames contain both the differences from the previous and the following frame, as illustrated on Figure 3.10.

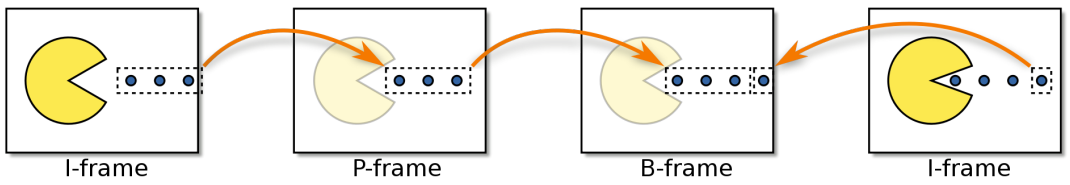

Figure 3.10: A sequence of Intra-coded, Predicted and Bi-predicted frames in a compressed video sequence. (Source: Petteri Aimonen, 2009 [23])

The last mentioned way of redundancy was spacial. This sort of redundancy is reduced by

taking advantage of coding techniques such as the DCT - Discrete Cosine Transform - which converts blocks of pixels into their frequency components. The great advantage of using the DCT comes when the HSV is considered - our visual system presents less sensitivity to variations in high frequencies, which enables the disregard for these components without greater perceived quality loss[24]. A typical chain of coding using the DCT consists of obtaining the DCT coefficients, performing quantification with weighted tables according to the HVS and finalizing by running VLC (variable length coding) and/or similar techniques.

3.4.1 MPEG-C part 3

The Moving Picture Experts Group (MPEG) specified MPEG-C part 3 in 2007 as a way of coding V+D streams. This standard is backwards compatible with 2D (video and depth being coded separately), is display and capture system independent, has a good compression efficiency as it creates low overhead and the global depth range can be user-controlled. The depth information can be obtained using extraction algorithms from 2D video, capturing scenes using RGB and infrared cameras or using stereo cameras. The depth is coded typically using 8-bit and it is compressed like a regular luminance map. Both the video and the depth streams are multiplexed together for transmission, granting this algorithm a somewhat low complexity. Additional view generation can be achieved within a limited angle via 3D warp, as the lack of other real camera angles create occlusion. A view generation example is shown in Figure 3.11.

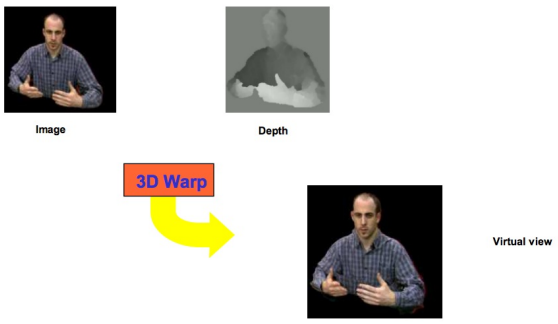

Figure 3.11: V+D additional view generation (Source: Aljoscha Smolić, 2007 [25])

The issue with occlusion areas is present in one other limitation, created by great depth values - which is not worrisome as the HVS does not react well to great depth changes (as discussed in section 3.1). This solution is able of 2D, 3D and, in a limited way, multiview 3D representation[26].

3.4.2 H.264/AVC

H.264/MPEG-4 Part 10 or H.264/AVC (Advanced Video Coding) is a codec standard developed by the Video Coding Experts Group (VCEG) and the MPEG. This codec is block-oriented and does motion-compensation, providing the necessary efficiency and flexibility to become the most

accepted standard for video coding [27]. Variable block-size allows an improved region determination during motion compensation, which is sublimed by the usage of multiple motion vectors on each macroblock and quarter-pixel precision. With the potential of reducing 50% of the bit rate when comparing to previous standards [28], H.264/AVC has improved the establishing chances of IPTV and HDTV. Switching I and Switching P frames written in the standard makes stream-switching easier, although major AVC implementations do not use them. As seen previously on section 3.3, the most simple approach to encode several views can be the H.264/AVC simulcast, where video streams are individually encoded. This allows for a separation of views with no great complexity.

One of the main advantages of H.264 is the use of the Network Abstraction Layer (NAL). This layer creates two types of units: the VCL (Video Coding Layer) and the non-VCL. The VCL NAL units encapsulate the video itself, partitioning the video data and allowing for the usage of the same video syntax regardless of the network and identify which picture parameter set is relevant to that unit. Non-VCL NAL units are used for transmitting extra information and can be divided into Parameter Sets (Sequence - SPS and Picture - PPS) and Supplemental Enhancement Information (SEI). PPS contain information that several video packets have in common and that is rarely changed, such as the video format, while SPS form the sequence-level header information, necessary for ordering and timing. SEI units provide not only timing but additional information to improve the decoding or modifications to the bit stream, for example. Grouping consecutive NAL (VCL and non-VCL) units associated with a specific decoded picture creates an Access Unit, which contains not only that picture's macroblocks but also redundant slices to improve error-resilience. The macroblocks in each picture are grouped into slices, which can be decoded individually. The group of sequential access units and its associated parameters form the bitstream [29].

The division into macroblocks and independently decodable slices offer a great deal of flexibility when accessing the bitstream. This flexibility, the robustness in network transmission and coding efficiency are the main advantages of using H.264.

3.4.3 H.264/MVC

As of the eleventh version of the H.264 standard - in 2009 - Multiview Coding support is extended, exploring inter-view redundancy, greatly improving its efficiency for MVV (H.264/MVC). Inter-view redundancy, as shown in Figure 3.7, uses a main bit-stream - meaning one independent view - to act as the base in which the other views are coded. This allows not only the decoding of MVC by single-view decoders but it also drastically reduces the amount of required data for the transmission of Multiview Video. The base view stream is coded in NAL units compatible with 2D decoders. The other views' streams are encapsulated using MVC NAL extension units. These units are prefixed with flags that can be used to identify each view, define their temporal hierarchy and if they are used for inter-view reference, among other parameters[30]. The structure of an MVC stream can be observed in Figure 3.12.

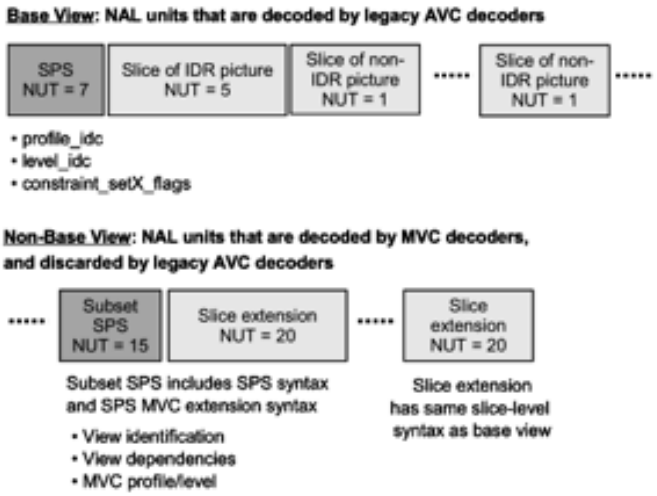

Figure 3.12: Structure of an MVC bitstream including NAL units that are associated with a base view and NAL units that are associated with a non-base view. NAL Unit Types (NUT) indicators are used to distinguish different types of data that are carried in the bit stream.
(Source: Vetro, Wiegand and Sullivan, 2011 [30])

H.264/MVC enables an efficient way of coding both 3D and multiview 3D video, by using inter-view prediction along with all the methods H.264/AVC used, while maintaining backward compatibility.

3.4.4 H.265/HEVC

High Efficiency Video Coding (HEVC) or H.265 is a codec standard published by the MPEG and the VCEG. Developed as the successor of H.264/AVC it addresses all of its current applications with special focus on the new resolutions (4K - Ultra HD, twice the horizontal and vertical resolution of 1080p - and higher) and on parallel processing architectures. With a more complex implementation than H.264/AVC, HEVC can achieve up to 50% bit-rate reduction maintaining perceptual quality [31]. Encoder efficiency for MVV can be increased if HEVC uses depth coding. An extension of HEVC supporting multiple views, depth coding and compensated prediction was proposed in May 2012. For 2 and 3 views, this approach achieved 40 and 50% bit-rate decrease, respectively, when comparing with simulcast [32]. HEVC encoders can adapt parameters such as the computational complexity and the compression rate to the application requirements. Motion compensation is done much more efficiently as its predecessor by using 33 directional modes versus the 8 used. Replacing macroblocks with Coding Tree Units, blocks of 64x64, 32x32, or 16x16 pixels based on its content, also increases efficiency. As of June 7th, the HEVC standard was formally published on the ITU-T website [33]. Despite all the promise it shows for being the next big video standard, the successor of H.264, its infancy status determined its impossibility of usage in this Dissertation.

3.5 Audio

Out of the main focus of this Dissertation, audio should not be, however, forgotten. When viewer immersion is the main goal, audio plays an important role. 3D audio's main objective is giving the feeling of space and directionality to the video's sound. Several 3D audio technologies are available (Vector Base Amplitude Panning, binaural techniques, Wave Field Synthesis, Ambisonics...). According to André, C in [34], only binaural techniques through headphones can achieve the desired immersion.

3.6 Network

Even with all the coding alternatives, transmitting 3D HD Multiview Video requires a lot of bandwidth and puts strain on the network infrastructures. The transport layer cannot be, therefore, overlooked. Choosing the right way of transmitting video and data can determine the success or failure of any implemented system. First, the possible network architectures are discussed, followed by how the transport is made in MPEG codecs, including H.264, by using MPEG-TS. In the last sub-section, an idea of how network congestion can be dealt with by adapting the content is given.

3.6.1 Client-Server networks

Current IPTV broadcast architectures are in the vast majority client-server based, meaning centralized servers distribute video streams directly to their clients. This traditional solution requires high processing power, great storage capacity and high bandwidth connection on the server end. Being centralized means it is more vulnerable to failures, which is why some content distributors use a mesh of servers, despite it being a much more expensive solution. Client-Server architectures enable a greater control on the content distributor side, facilitating the implementation of DRM and overall content supervision.

3.6.2 P2P networks

An interesting work - "Adaptive Multi-view Video Streaming over P2P Networks Considering Quality of Experience" [35] - studies the possibility of using Peer-to-Peer (P2P) to transmit MVV (H.264/AVC, not MVC - as the author considered that MVC's encoding gains could become marginal depending on number of views and capture factors). This work addresses the main two challenges of P2P as a broadcast medium - network resources availability and scalability. So that Quality of Experience (QoE) is assured, the risk of IP failures and latency should be minimized. P2P networks distribute the network load over its peers, providing a faster way of content sharing. Often associated with the distribution of illegal contents in the eyes of DRM holders, P2P's efficiency could make it the next way of content broadcast.

3.6.3 MPEG-TS

MPEG-Transport Stream is a standard specified in MPEG-2 Part 1, Systems in 1995. This standard is used in various network environments and works by encapsulating and multiplexing the Packetized Elementary Stream packets, supporting multiple time bases, and other necessary information such as error correction and synchronization bits. Based on a layered approach, the video is decomposed in packets for transmission, enabling compatibility with IP-based broadcasting systems, such as IPTV. MPEG-TS groups the video slices (I, B, P-frames) into Group of Pictures (GOP), allowing random access to the stream. Grouping the GOPs into a data stream creates an Elementary Stream, which is then divided into packets, forming the Packetized Elementary Stream (PES). One of the advantages of MPEG-TS is allowing for the multiplexing of video streams with data, such as program specific information. This is specially relevant for broadcasting, providing a way of sending program association tables and program map tables, which list and present program information accordingly[36].

3.6.4 Content Adaptation

Other possibility of dealing with the high-bandwidth requirements in Client-Server setups is using resource-oriented content adaptation. By analyzing network traffic the system can adapt the content to the network. Scalability and graceful degradation are on the strong features of the H.264/MPEG-4 AVC standard, made possible by their profiles and levels, which adjust the quality (Peak Signal-to-noise ratio adjustments, for instance), temporal resolution (number of frames per second), spatial resolution and number of views being sent to the viewers. Catering to the reality of each viewer's connection avoids sending excessive data, which minimizes lag, jitter and improves overall user experience[37].

3.7 Conclusion

After doing extensive research on these subjects, a global perception of the State of the Art of 3D technologies was developed, which allowed for a better comprehension of how a 3D system works and improved the solution-seeking skills inherent to the work developed. For this Dissertation, the ideal 3D display would be one that allowed auto-stereoscopy. Due to the cost of multi-projector solutions and because of the early development stage volumetric displays are still in, an Autostereoscopic Lenticular Display would be the perfect candidate for testing the developed work.

Considering all 3D formats available, the chosen format would naturally be one which has support for multiview video, like the MVV format. For the compression standard, MPEG-C part 3 seemed to be a good solution in terms of efficiency, but its limits when it comes to Multiview Video proved otherwise. H.265 shows great promise due to its coding efficiency and functionality (covering all of H.264 ones). Unfortunately it is still in its early stage, having only been published as a standard when the dissertation work was at its final phase.

Due to its efficiency, Multiview Video Support and worldwide adoption, H.264 was chosen.

Chapter 4

Media Server

In this chapter, a description of the Sirannon Project, where the chosen Media Server comes from is given, followed by the components and modules used in this Dissertation and what media files were used.

4.1 The Sirannon Project

Sirannon is a modular multimedia streamer and receiver developed by Alexis Rombaut that allows the user to configure parameters of the modules used in the s2ystem. It is written in C++ and supports HTML5 - WebM (open media format for the web), Apple Live HTTP Streaming, H.264/SVC and packet loss modeling.

Running on Ubuntu 12, Sirannon also supports several audio and video codecs, whose containers can be MPEG-2 PS, MPEG-2 TS, AVI, MOV/MP4, MKV, WebM or none (raw). In terms of transmission protocols it supports the most used ones: RTMP, RTMPT, RTSP / RTP / UDP, HTTP, UDP and TCP. Sirannon is free, open source, cross-platform and a part of the reference toolchain by the Video Quality Experts Group. Documentation and installer are available at the project's webpage - `http://sirannon.atlantis.ugent.be/`.

4.2 Components and Modules

Setting up Sirannon usually involves linking several modules in blocks, as shown in Figure 4.1. These connections can channel audio and/or video according to the route number established in the module and in the connector (usually 100 for video, 200 for audio).

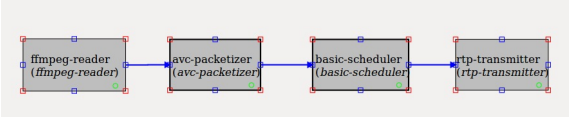

Figure 4.1: Example of connected modules in Sirannon

In this example, a media reader (ffmpeg-reader) is connected to a packetizer module which distributes the data in AVC-specific packets. The basic-scheduler module ensures the packets are in the right order and the RTP-Transmitter streams the packets to their destination (using RTP). Each individual module can be modified by altering the code in its individual file and then recompiling Sirannon. Their individual preferences can be adjusted via a side preference pane (Figure 4.2).

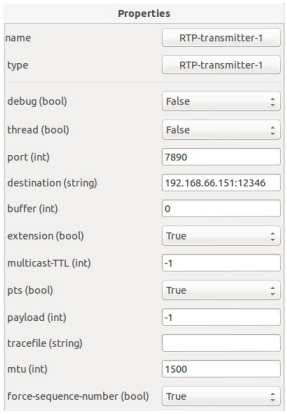

Figure 4.2: Example of a module's preference pane

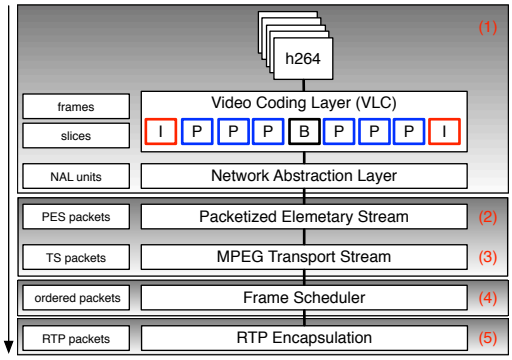

Figure 4.3: Video Chain: from H.264 to transmission packets

Figure 4.3 shows a transmission chain from the raw H.264 video file to the RTP packets used in the transmission. Applying this chain to Sirannon's modules, the following correspondence is found:

1. a file reader module: avc-reader, specific to raw H.264, which reads the file and generates media packets consisting of one NAL-unit (Network Abstraction Layer), usually the bits of Macro-blocks in a slice as payload. Or an FFMPEG, a module based on the multiformat and multicodec open-source video swiss-army-knife (encoder, decoder, mux, demux, ...) [38].

2. a packetizer module, which groups the packets of one frame into one PES-packet (Packetized Elementary Stream);

3. an MPEG-TS (Transport System) module multiplexes the audio and video together;

4. a frame-scheduler module, which ensures the sending of the packets that belong to one frame is transmitted with a temporal distribution, preventing them to be sent in a single burst;

5. a transmitter module, which encapsulates with necessary UDP/RTP headers and automatically generates the RTCP packets, transmitting them afterwards to the provided destination;

6. a sink, required by Sirannon to finalize the chain, terminating the program after receiving the last packet sent by RTP-Transmitter.

4.2.1 Two view-pairs configuration for Multiview

As a first approach, only two view-pairs are transmitted. Each view-pair consists of two video files, a left and a right one, in pure H.264 format. The modular configuration is shown in Figure 4.4:

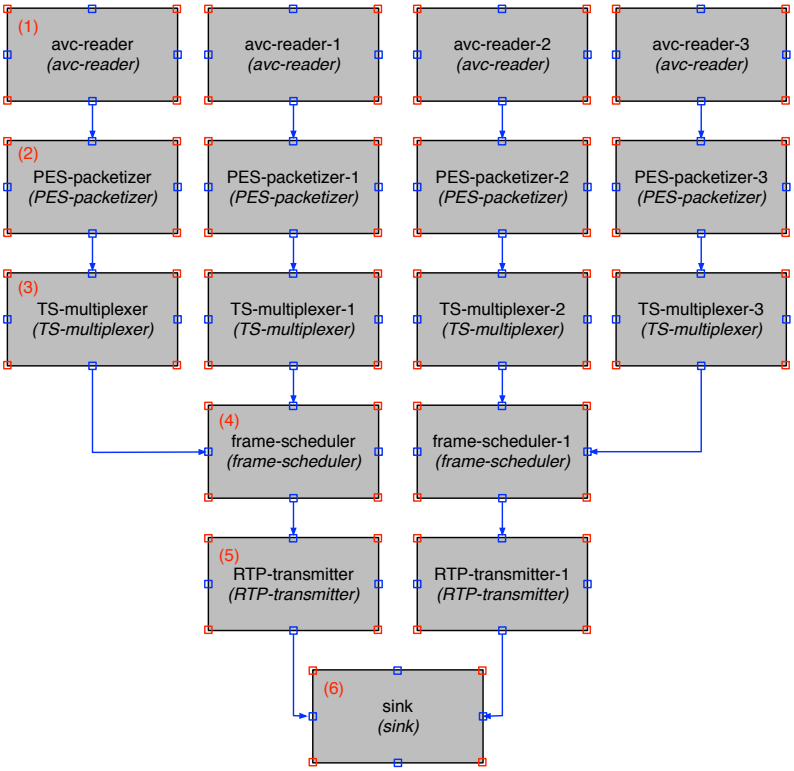

Figure 4.4: Two view-pairs configuration in Sirannon

4.2.2 Three view-pairs configuration for Multiview

The required modifications of the configuration described on 4.2.1 to add a third view-pair are shown in Figure 4.5.

Besides the addition of a new view pair, this model is slightly different than the previous one. The reader module is an FFMPEG one - this is due to the short length of the pure H.264 video files used in avc-reader, not long enough for extensive transition testings. On this reader module, two minute long mp4 files were used. The FFMPEG module reads the video file and generates fixed length packets for every frame. But the great change in this configuration appears at the frame-scheduler level, where all the views converge. By doing so, the view-selection decision is passed on to the transmitter module, which allows for an easier control of transmitted views.

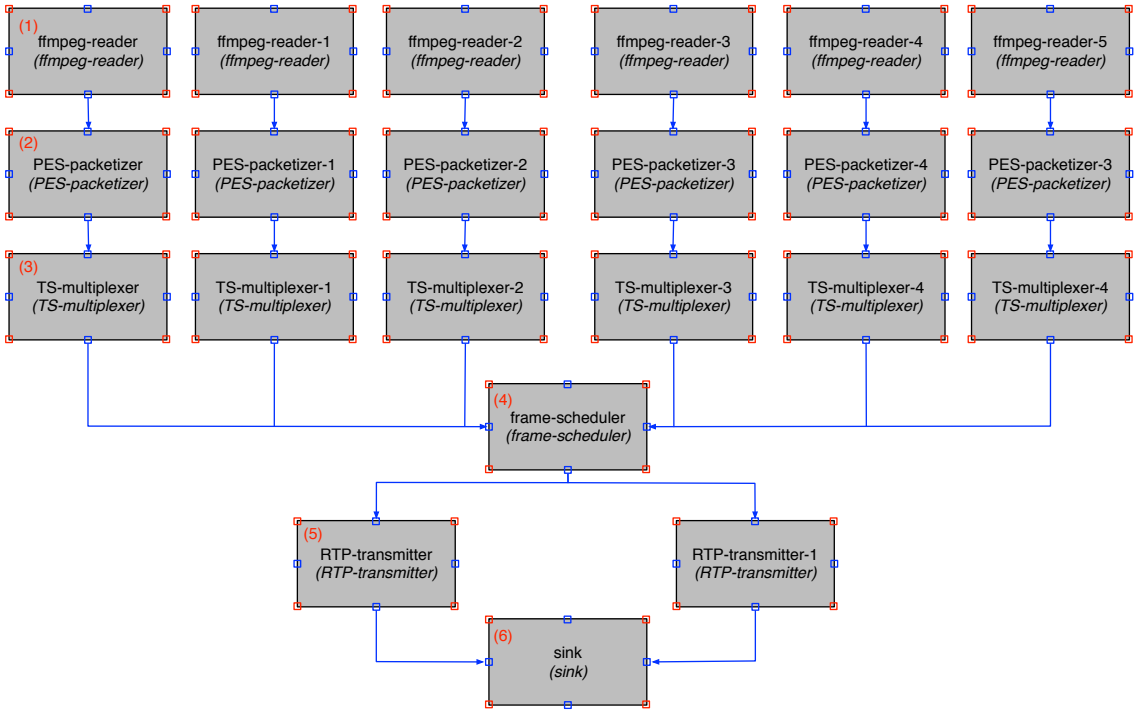

Figure 4.5: Three view-pairs configuration in Sirannon

4.2.3 View Change

The selection of the currently active view is done based on the headtracking information sent by the client. For the two-view-pair case, if the yaw value received changes signal, the transmitted view changes (negative yaw for the left view-pair and positive for the right). The view-pair change can occur anywhere in the stream, which means that a change can happen outside the GOP transition, which creates artifacts until the new GOP with a new I-frame arrives. This was a required sacrifice in order to show how responsive this system can be without network overloads due to multiple full resolution streams.

4.3 Media Files

The media files used in 4.2.1 were from the JMVC [39] (Joint Multiview Video Coding) software for the Multiview Video Coding (MVC) project of the Joint Video Team (JVT) of the ISO/IEC Moving Pictures Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG). These files were in yuv format and were later converted into raw H.264 with the .264 extension.

Chapter 5

Client Interface

In this chapter, a description of the Client Interface's features is given, followed by a detailed characterization of its main modules. The last section gives information on how the Client-Server communication is conducted.

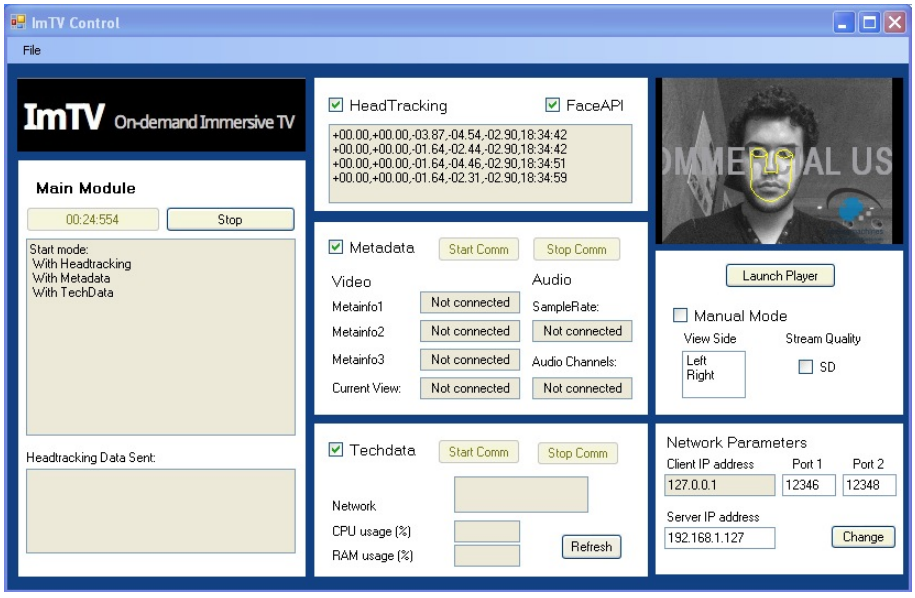

Figure 5.1: Client Interface

5.1 Features

The Client Interface is a multi-feature program for the Windows^{TM} operating system to be run on the Client side. This software's main features include processing the headtracking information sent by faceAPI, measuring and transmitting the Client operating system's performance, displaying the video and audio's metadata, launching an external player and performing manual view and quality

change. Its appearance is as shown in Figure 5.1. This program is capable of saving settings such as the Server IP address, the ports used and the location of the FaceAPI executable between runs.

5.2 Main Modules

Separating the software in modules was considered a better way of implementing the previously mentioned features in a practical and organized way. This separation allowed independent threads running for each module, improving performance. The following main modules were created.

5.2.1 Headtracking

The Headtracking is done by launching on a separate thread an external application - Socket.exe - a program developed by the faceAPI team which processes the headtracking information and outputs it to the application console, which is re-routed to the User Interface Application. This application uses faceAPI and through a web-camera measures the position of the viewer's head - translation (X, Y, Z) in centimeters, neck roll, pitch and yaw in degrees and the confidence level (conf_value) which goes from 0 to 1, as shown in Figure 5.2.

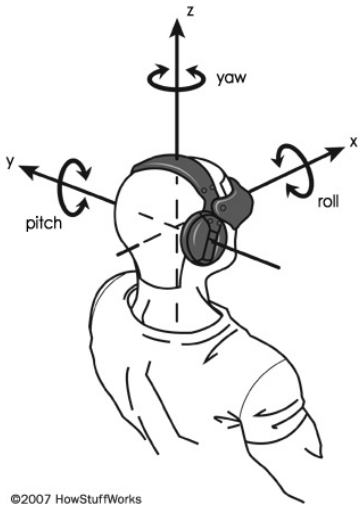

Figure 5.2: Head position definition: translation, pitch, yaw, and roll
(Source: How Stuff Works, 2007[40])

The output of Socket.exe comes as the following string:

"Head Pose: pos(X,Y,Z) rot(pitch, yaw, roll) conf +conf_value"

The original string is modified to use less bytes and to prevent issues that characters such as parenthesis and spaces could originate. The values are transmitted to the Server exactly in the format "+00.00" (a plus or minus sign and the decimal value) separated by commas. A timestamp is added at the end for control purposes. The final string is transmitted to the UDP Port 5007 in the format:

"+X,+Y,+Z,+pitch,+yaw,+roll,timestamp"

"-00.04,+00.00,-08.74,+06.81,+02.03,17:58:52" as an example. The zero value is sent with a positive sign, merely as a convention.

In order to prevent sending unnecessary packets, the headtracking module only sends a string once a threshold of head movement is detected. This threshold was defined as more than two centimeters (in the case of translation) or more than two degrees (in the case of rotation) by experimenting which amount of output lines would be empirically acceptable.

5.2.2 Metadata

For the video, the Server reads and transmits the following metadata:

- Resolution - The spacial dimension of the video, in pixels (width x height);
- Bitrate - The number of bits present in the video per second (bps);
- Framerate - The amount of video frames per second (fps);
- Timebase - used by the codec as a way of referencing time, it is the timescale as the numerator and the number of units in a tick (usually meaning the frame rate). as denominator (180.000/50, for instance);
- Inc - the result of the ratio explicit in Timebase (3600 as an example).

This module receives the different kinds of metadata both of video and audio and displays it to the user.

The following table shows the kind of video metadata the Server retrieves from the read files and the Client program is prepared to receive (adapting to different kinds can be easily coded):

	Resolution	Bitrate	Framerate	Timebase	Inc
MP4	X	X	X		
H264			X	X	X

Table 5.1: Video metadata

For the audio, the following metadata is used:

- Sample Rate - The number of audio samples per second in Hertz (Hz).
- Channels - The number of audio channels used (usually two).

The metadata reception is done on UDP port 5008 and only the received fields are shown, uncluttering the interface of unnecessary information. The Metadata messages are of the following formatting: "metadata_item:item_value", "Resolution:1920x1080", for instance.

In addition to the metadata read from the media files, the Metadata module also transmits the current view, using the following format: "Current_View:view", "Current_View:Right" as an example.

5.2.3 Techdata

The Techdata - technical data - module measures the Client machine and the network performance, sending it to the Server using UDP port 5009. The following data is collected:

- CPU usage (%) - Uses the .NET Framework PerformanceCounter library to measure the usage of the CPU by taking samples of the running processes.
- Memory usage (%) - Uses the same library as the CPU usage counter to measure the total used RAM and the total amount of memory installed in the system. The ratio between the two gives the memory usage.
- Lost packets - Uses the System.Net.NetworkInformation library to send a predefined number (10) of Internet Control Message Protocol (ICMP) echo messages - ping messages - to the Server.
- Loss percentage - Calculates the loss percentage using the ratio between the number of lost echo messages and the total sent.

Each Techdata item is sent at a time in the following format:

"RAMusage CPUusage LostPackets LossPercentage"

"50 23 1 10", as an example. This string corresponds to a 50% RAM usage, a 23% CPU usage, one lost packet and 10% loss of packets (one lost packet in 10 sent).

5.2.4 3D Player

The chosen player in Chapter 2, sub-section 2.4.4, was Bino. Since Bino is an on-going project, there were some issues with it running on Windows™ (crashes in particular) that did not happen on its Ubuntu version, where it managed to play a 3D video consisting of two separately sent view-pairs and output them as a stereoscopic video. In order to test view transitions on Windows™, VLC player was used by launching two instances of this player each with the port corresponding to the side of the view pair transmitted.

As this Gent University project continues, issues like crashes on Windows™ are hoped to be fixed.

5.2.4.1 Bino configuration

On Bino, one can choose both the input (Figure 5.3) and the output (Figure 5.4 layout).

Besides a 2D mode, the input comes in view-pair format and can be:

- single stream:
  - alternating - one view in each frame alternating;

- two streams:
  - top-bottom - each frame has two full views spatially distributed,
  - top-bottom-half - a mix of top-bottom and alternating,
  - left-right - like top-bottom but the views are distributed spatially side by side
  - left-right-half - a mix of left-right and alternating
  - even-odd - each view-pair is interlaced, either by rows or columns.

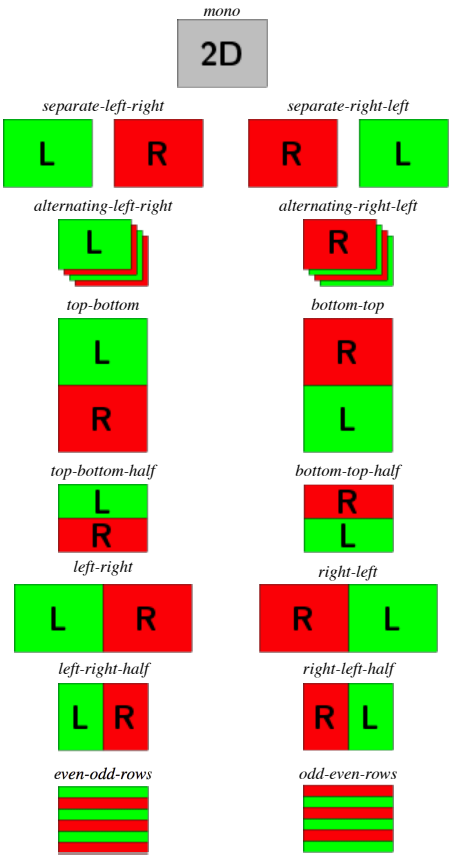

Figure 5.3: Bino input layouts
(Source: Bino Manual, 2011 [41])

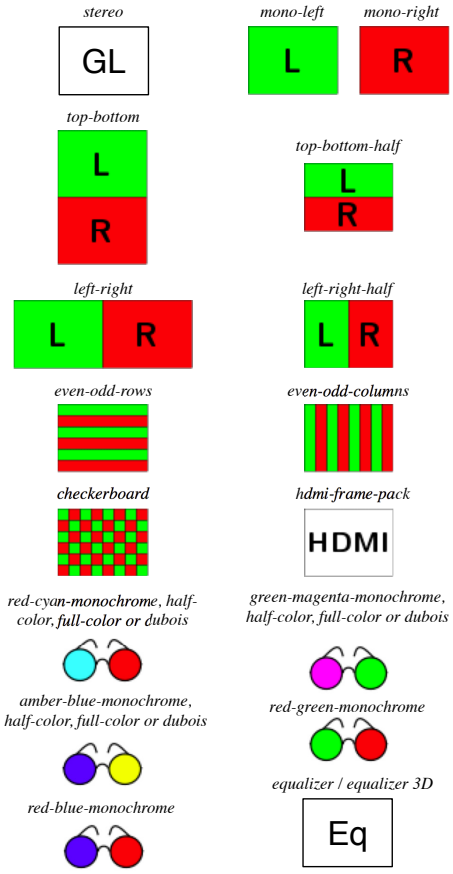

Figure 5.4: Bino output layouts
(Source: Bino Manual, 2011 [41])

Bino supports various output modes: OpenGL quad-buffer stereo (specific mode for 3D rendering in OpenGL), mono - by selecting which one of the views it displays, top-bottom, top-bottom-half, left-right, left-right-half, even-odd (all previously mentioned were described in the input part), checkerboard (views displayed in a checkerboard pattern), HDMI frame packing mode (for compatibility with HDMI 3D mode), numerous anaglyph 3D modes and a multi-display OpenGL via Equalizer with a 2D or a 3D canvas setup.

5.2.5 Manual Modes

The adaptation to the network conditions done by the Server allows for an easy way of implementing a choice of video quality. It is set on the Server that if the loss percentage techdata parameter is greater than 10%, the Server sends the lower quality stream (SD - Standard Definition). Through forcing a value of 20%, as an example, the Client can force the change from an HD to an SD stream.

Another manual mode consists on choosing which view pair is currently selected. This mode disables the sending of the real headtracking data and instead sends dummy versions - "+00.00,-+00.00,+00.00,-02.00,+00.00,timestamp" for changing to the view pair at its left and "+00.00,-+00.00,+00.00,+02.00,+00.00,timestamp" for changing to the view pair at its right.

5.3 Server Communication

Knowing that the UDP protocol is connection-less - no guarantee of delivery, ordering or error proofing, the communication with the Server was based on a ping-pong type communication (transmission-response). This protocol was also chosen considering that only the the latest information received is important and that the UDP protocol simplifies the implementation of the network channel between Server and Client.

Each module communication is done in its own UDP channel and the exchanged messages are different in each module.

The starting and ending communication messages were defined as having the following configuration:

STARTING_COMMUNICATIONS_WITH_SERVER_SOCKETS for initiating the connection and ENDING_COMMUNICATIONS_WITH_SERVER_SOCKETS for terminating it.

The following Figures (5.5, 5.6 and 5.7) illustrate the communication protocol for each of the main modules.

5.3.1 Headtracking communication

The Server initializes the communication by sending the starting message. Until the Client replies, the Server keeps re-transmitting that message. Once the Client replies with the starting message, it starts sending the headtracking strings to the Server with the content mentioned in sub-section 5.2.1. Due to the importance of the headtracking data, the Server is required to reply with the exact received string. If this does not happen, the Client re-transmits until the Server does, so that the Headtracking data has a real impact on the view change, once a headtracking worthy of creating a view-change is detected it should be quickly transmitted. Having this in mind, and in order not to flood the network with packets, a 0.1 second delay was created between headtracking messages. To this delay time, both Server processing and network transmission times have to be accounted for. When a side (either the Client or the Server) wants to end the communication, the termination

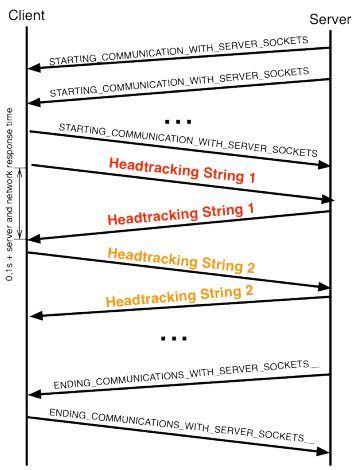

Figure 5.5: Headtracking communication protocol

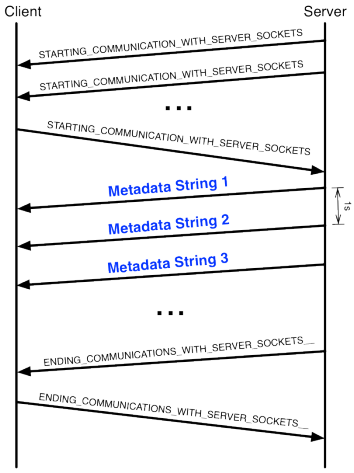

Figure 5.6: Metadata communication protocol

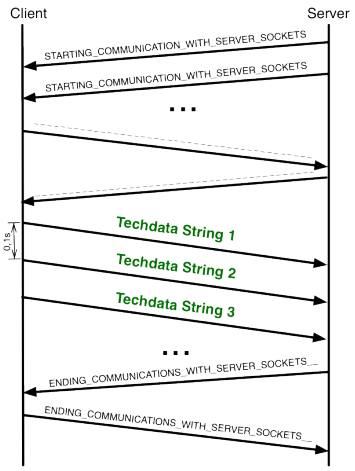

Figure 5.7: Techdata communication protocol

string is sent, after which the opposite side replies with the same message. Figure 5.5 illustrates the full protocol just described.

5.3.2 Metadata communication

After the Server and the Client initialize the connection as described on the Headtracking communication sub-section (5.3.1), the Server sends with a 1 second interval the Metadata string described in the sub-section 5.2.2. These messages are sent until all the metadata values are transmitted. This cadence of message transmission is slower than that in the Headtracking Communication due the data being low-priority. After every type of Metadata is transmitted, the Server starts a new cycle of transmitting values as a way of keeping the communication channel opened and the data in the Client updated, in case the stream changes, for instance. This protocol is shown in Figure 5.6.

5.3.3 Techdata communication

In this module, the Client has to know if the Server has gone through the initiation phase but there is no need for data validation or assurance of reception - lower data priority than the Headtracking. Taking this into consideration, changes were made in the initiation protocol. The Server sends the starting message and the Client replies with an underscore-filled message of exactly 43 characters (same length as the starting and ending messages). Once the Server replies with that exact message, and not the typical starting message, the Client will start sending the Techdata strings - as described in subsection 5.2.3 - using a 0,1 second interval. This interval was chosen due to the fact that the Techdata's information can affect what stream the Server transmits. Figure 5.7 illustrates the Techdata communication protocol.

Chapter 6

Tests and Results

As a way of validating the implemented system, tests were conducted. Although purely subjective tests described in section 6.1, some objective testing was carried out using Iperf, described in section 6.2.

6.1 Subjective tests

The subjective testing consisted in transmitting 3D video to the client's machine. Due to the unavailability of a stereoscopic display, the video was either played on Bino, in the case of raw h.264, or on VLC.

On Bino the input chosen mode was separate left-right and the output even-odd-rows for a sense of stereoscopy, the left-right was also chosen when there was the need to analyze the integrity of the received video (for a description of all of these layouts, please refer to sub-section 5.2.4.1 on Chapter 5.

On VLC, two program instances were kept running at the same time, each programmed to a different RTP stream, the main stream would display one view of the main view-pair, while the other stream would display the degraded one.

The subjective testing done was satisfactory in the sense that it was possible to receive 3D video in both players, visualize transitions induced by both the performed headtracking and the manual mode, receive updated metadata on the client (including the current view when it changes), send Techdata and experience its effects on video quality (by activating SD manual mode). A wireless Connection (IEEE 802.11n) was used and no issues arose from using it.

6.2 Tests with the Iperf

Iperf (as mentioned in Chapter 2, sub-sub-chapter 2.4.3.2) is a network congestion analyzer able to provoke congestion. The procedure for testing with Iperf consisted in doing both unidirectional and bidirectional tests in three variations: **free** mode - no Client-Server communication, **AVC** mode - transmitting raw H.264/MPEG-4 AVC 3D video streams (two view pairs) and **MP4** mode

- transmitting three view-pairs by streaming MPEG-4 MP4 contained files. The results are shown in Table 6.1.

		Bandwidth (Mbit/s)		Jitter (seconds)		Loss Percentage	
		Client	Server	Client	Server	Client	Server
Free	Unidirectional	631		0,058		13%	
	Bidirectional	403	367	0,041	0,116	0,11%	9%
AVC	Unidirectional	517		0,135		3,1%	
	Bidirectional	611	605	0	0,098	0%	0,89%
MP4	Unidirectional	443		0,107		1,7%	
	Bidirectional	497	363	0,035	0,062	1,7%	15%

Table 6.1: Network load tests

These tests were performed using a Gigabit Ethernet connection (using CAT5 cables, nonetheless), instead of the wireless connection mentioned before. This approach was chosen to avoid the wireless signal fluctuations that could affect the test results. Through the results on the table above, some inferences were made:

- Due to the short duration and the low resolution (640x480 pixels) of the raw AVC video, the effect on the network was minimized.
- When using the MP4 videos, the effect on the network was more noticeable, although no real issues came from it, as one can see in both the jitter and the loss of packets.
- Considering nowadays' fiber-optic network infrastructures proliferation, particularly fiber-to-the-home (FTTH), users with bandwidths of 30, 50 or 100 Mbits/second are typical in Portugal [3], transmitting multiple HD streams would be a non-issue, seeing that the amount of data transmitted along side the video can be considered residual.

Chapter 7

Conclusions

In this chapter, some remarks are made regarding the encountered difficulties (in 7.1) and how the development of this project can be continued using the work done so far (in 7.2). A final remark is done in the last section, 7.3.

7.1 Difficulties

This Dissertation work did not come without difficulties, some expected, but others completely unforeseen.

The first real difficulty was felt during the phase of familiarization with the project. Grasping concepts such as 3D multiview video seem now trivial when compared with understanding Sirannon - the media Server - and its multiple 1000-line-long C++ files. Some unexpected behaviors intrinsic to a still-in-development program added the necessity to allocate more time to explore Sirannon. One somewhat disappointing problem of Sirannon was its limitation in number of streams it can transmit at the same time, being this limit reached when trying to stream four view-pairs - it is important to note that this limitation was not due to the machine running Sirannon (the system had still vast available resources), but to Sirannon coding.

When trying to modify and possibly correct Bino - the 3D player - there was the need to install a cross-compiler. The most current version of Bino - 1.4.2 - had issues when opening network streams in Windows™, which resulted in hanging and then crashing, which did not happened in the Ubuntu version, at least not every time it ran. The recommended compiler to produce a Windows™ version, MinGW, proved to be virtually impossible to run in both Ubuntu and Windows™ due to the difficulty in finding the necessary libraries, even though they were installed and correctly referenced. More than the allocated time to this issue was spent, which forced a change of orientation towards the most pressing features, such as programming the Client interface.

Finding multiview-3D videos in the AVC or MVC video format was not trivial, and the files found had really short duration (seconds), which increased the difficulty in performing view transitions. Due to Sirannon's issues with graceful termination of a video stream, looping these videos

was not an option.

7.2 Future Work

Due to the complexity of the work at hand, it was not possible — nor was under the scope of this dissertation work — to accomplish every single feature thought out. The work can be developed further in multiple fronts.

One possibility is exploring the limitations observed in the media Server: the apparent maximum of number of view-pairs it supports, the lack of a "graceful" stream termination - issues with the Sink Module not working properly and also implementing DRM.

Other possibility pertains to the 3D player: fixing Bino in Windows^{TM} can be considered somewhat of a priority, especially if an auto-stereoscopic display becomes available for the project. In the case this is not at all possible, trying to program VLC to display stereoscopic content would be the next step.

A radical change of paradigm in the way view-change is done can be achieved by placing the focus on the Client: whenever a view-change is detected, the Client would automatically change the current stream using the low quality version and then make the switch to the new full quality stream when the Server makes the change.

Changing the video files to ones coded using the new HEVC, once this standard is completed, would be extremely valuable to attain the expected reductions in video size and exchange network data.

Being able to take advantage of 3D audio would also be an interesting idea. Using 3D audio would create a greater sense of immersion and therefore improve the user experience.

A final improvement presents a greater challenge in terms of implementation and research: exploring ways of allowing more than one viewer in one Client at a time. This would be done by initially researching a multi-face detection software and then figuring out how can a person be selected in that program as the main or commanding one (being this person the one whose facial movements affect view-change).

7.3 Final Remarks

Throughout the duration of this Dissertation, a strong grasp of what the 3D area is and what can be done in adapting 3D content to a multiview paradigm was achieved. Watching auto-stereoscopic multiview videos without the need of wearing glasses or other kind of contraption is the right way of giving this new technology a change to thrive in the consumer market.

In spite of the difficulties described previously, this Dissertation allowed for some important and incredibly useful skills to be developed and hopefully helped pushing forward the barrier of knowledge in this area.

References

[1] Toby Miller. *Television Studies: the basics*. Routledge, 2009.

[2] Christopher Nickson. The future of television and hdtv. *Digital Trends Online*, March 2009. URL: http://www.digitaltrends.com/features/the-future-of-television-and-hdtv/.

[3] ANACOM. Mercados grossistas de acesso à infraestrutura de rede num local fixo e de acesso em banda larga. URL: http://www.anacom.pt/streaming/doc_consulta_MercadGrossista4_5.pdf?contentId=1116435&field=ATTACHED_FILE [last accessed Sunday, 16 June 2013].

[4] 3d tv share of global lcd tv panel shipments from the 2nd quarter of 2011 to the 4th quarter of 2012 [online]. March 2012. URL: http://www.statista.com/statistics/220081/global-market-share-of-3d-tv-panels/ [last accessed Monday, 27 May 2013].

[5] Display search website [online]. URL: http://www.displaysearch.com [last accessed Monday, 27 May 2013].

[6] Fnac 3d displays. URL: http://www.fnac.pt/imagem-e-som/TV-3D/Televisores/s21075?bl=HGAChead [last accessed 12th February 2013].

[7] Wireshark. Wireshark - about. URL: http://www.wireshark.org/about.html [last accessed Sunday, 16 June 2013].

[8] Iperf. Iperf website. URL: http://iperf.sourceforge.net/ [last accessed Sunday, 16 June 2013].

[9] Gabriele Longhi. State of the art 3d technologies and mvv end to end system design. Master's thesis, Università Degli Studi di Padova, Facoltà di Ingegneria, April 2010.

[10] W.J. Tam, F. Speranza, S. Yano, K. Shimono, and H. Ono. Stereoscopic 3d-tv: Visual comfort. *Broadcasting, IEEE Transactions on*, 57(2):335 –346, june 2011. doi:10.1109/TBC.2011.2125070.

[11] Sumio Yano, Masaki Emoto, and Tetsuo Mitsuhashi. Two factors in visual fatigue caused by stereoscopic hdtv images. *Displays*, 25(4):141 – 150, 2004. URL: http://www.sciencedirect.com/science/article/pii/S0141938204000678, doi:10.1016/j.displa.2004.09.002.

[12] Y. Nojiri, H. Yamanoue, S. Ide, S. Yano, and F. Okana. Parallax distribution and visual comfort on stereoscopic hdtv. In *Proc. IBC*, pages 373–380, 2006.

[13] J. Konrad and M. Halle. 3-d displays and signal processing. *Signal Processing Magazine, IEEE*, 24(6):97 –111, nov. 2007. doi:10.1109/MSP.2007.905706.

[14] D. McAllister. Display technology: Stereo & 3d display technologies. *North Carolina State University*, 2005. URL: http://research.csc.ncsu.edu/stereographics/wiley.pdf.

[15] N.A. Dodgson. Autostereoscopic 3d displays. *Computer*, 38(8):31 – 36, aug. 2005. doi:10.1109/MC.2005.252.

[16] Jonathan Mather. Parallax barrier cross section, licensed under creative commons - http://creativecommons.org/licenses/by-sa/3.0/deed.en. URL: http://en.wikipedia.org/wiki/File:ParallaxBarrierCrossSection.svg [last accessed Monday, 17 June 2013].

[17] W. Matusik and H. Pfister. 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 814–824. ACM, 2004.

[18] The Graphics Lab at the University of Southern California. Rendering for an interactive 360º light field display. URL: http://gl.ict.usc.edu/Research/3DDisplay/ [last accessed 1st February 2013].

[19] G. Saygili, C.G. Gurler, and A.M. Tekalp. Evaluation of asymmetric stereo video coding and rate scaling for adaptive 3d video streaming. *Broadcasting, IEEE Transactions on*, 57(2):593 –601, june 2011. doi:10.1109/TBC.2011.2131450.

[20] P. Merkle, A. Smolic, K. Muller, and T. Wiegand. Multi-view video plus depth representation and coding. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 1, pages I–201. IEEE, 2007.

[21] B. Bartczak, P. Vandewalle, O. Grau, G. Briand, J. Fournier, P. Kerbiriou, M. Murdoch, M. Muller, R. Goris, R. Koch, and R. van der Vleuten. Display-independent 3d-tv production and delivery using the layered depth video format. *Broadcasting, IEEE Transactions on*, 57(2):477 –490, june 2011. doi:10.1109/TBC.2011.2120790.

[22] Maria Teresa Andrade. Televisão digital e novos serviços, slides, capítulo 1. FEUP, 2011/2012.

[23] Petteri Aimonen. I p and b frames, released into the public domain. URL: http://en.wikipedia.org/wiki/File:I_P_and_B_frames.svg [last accessed Sunday, 22 June 2013].

[24] Vasudev Bhaskaran and Konstantinos Konstantinides. *Image and Video Compression Standars: Algorithms and Architectures*. Kluwer Academic Pub, 1997.

[25] Ing Aljoscha Smolić and HHI Fraunhofer. Compression for 3dtv-with special focus on mpeg standards. In *3DTV Conference '07, Kos Island*, 2007.

[26] A. Bourge, J. Gobert, and F. Bruls. Mpeg-c part 3: Enabling the introduction of video plus depth contents. In *Proc. of IEEE Workshop on Content Generation and Coding for 3D-television, Eindhoven, The Netherlands*, 2006.

[27] P. Merkle, K. Müller, and T. Wiegand. 3d video: acquisition, coding, and display. *Consumer Electronics, IEEE Transactions on*, 56(2):946 –950, may 2010. doi:10.1109/TCE.2010.5506024.

[28] G.J. Sullivan and T. Wiegand. Video compression - from concepts to the h.264/avc standard. *Proceedings of the IEEE*, 93(1):18 –31, jan. 2005. doi:10.1109/JPROC.2004.839617.

[29] Internet Engineering Task Force (IETF). Rtp payload format for h.264 video. URL: https://tools.ietf.org/html/rfc6184 [last accessed Saturday, 22 June 2013].

[30] A. Vetro, T. Wiegand, and G.J. Sullivan. Overview of the stereo and multiview video coding extensions of the h.264/mpeg-4 avc standard. *Proceedings of the IEEE*, 99(4):626–642, 2011. doi:10.1109/JPROC.2010.2098830.

[31] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Trans. Circuits and Systems for Video Tech*, 2012.

[32] H. Schwarz, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, D. Marpe, P. Merkle, K. Muller, H. Rhee, G. Tech, M. Winken, and T. Wiegand. 3d video coding using advanced prediction, depth modeling, and encoder control methods. In *Picture Coding Symposium (PCS), 2012*, pages 1 –4, may 2012. doi:10.1109/PCS.2012.6213271.

[33] ITU-T. H.265 : High efficiency video coding. URL: http://www.itu.int/rec/T-REC-H.265 [last accessed Sunday, 23 June 2013].

[34] C. André, J.-J. Embrechts, and G.V. Jacques. Adding 3d sound to 3d cinema: Identification and evaluation of different reproduction techniques. In *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, pages 130 –137, nov. 2010. doi:10.1109/ICALIP.2010.5684993.

[35] Saadet Sedef Savas, Ahmet Murat Tekalp, and Cihat Goktug Gurler. Adaptive multi-view video streaming over p2p networks considering quality of experience. In *Proceedings of the 2011 ACM workshop on Social and behavioural networked media access*, SBNMA '11, pages 53–58, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/2072627.2072643, doi:10.1145/2072627.2072643.

[36] ITU. H.222.0 : Information technology - generic coding of moving pictures and associated audio information: Systems. Technical Report 4.0, ITU, 06 2012. URL: http://www.itu.int/rec/T-REC-H.222.0 [last accessed Wednesday, 5 June 2013].

[37] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the h.264/avc standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1103 –1120, sept. 2007. doi:10.1109/TCSVT.2007.905532.

[38] FFMPEG. Ffmpeg - about. URL: http://www.ffmpeg.org/about.html [last accessed Friday, 7 June 2013].

[39] JMVC. Jmvc - about. URL: http://www.jmvc.org/info.jmvc [last accessed Monday, 17 June 2013].

[40] How Stuff Works. How virtual reality gear works. URL: http://electronics.howstuffworks.com/gadgets/other-gadgets/VR-gear6.htm [last accessed Monday, 17 June 2013].

[41] Bino. Bino - input and output layouts. URL: http://bino3d.org/doc/bino.html#Input-Layouts [last accessed Sunday, 16 June 2013].