FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Advancing Computational Biophysics with Virtual Reality

Afonso Bernardino da Silva Pinto

Master's Degree in Informatics and Computer Engineering

Supervisor: António Augusto de Sousa

July 03, 2020

© Afonso Bernardino da Silva Pinto, 2020

Advancing Computational Biophysics with Virtual Reality

Afonso Bernardino da Silva Pinto

Master's Degree in Informatics and Computer Engineering

July 03, 2020

Resumo

A visualização científica, conceito interdisciplinar entre ciência e computação gráfica, tem como objetivo permitir uma conceptualização visual no processo de pesquisa cientifica. A capacidade dos cientistas visualizarem resultados de computações e simulações é essencial para a aquisição de novos conhecimentos e, por conseguinte, para evolução do campo científico envolvente.

O método científico é, por natureza, um processo interativo e iterativo o que requer que ferramentas de auxílio à visualização gráfica sejam o mínimo limitantes possível no que diz respeito à interação com o ambiente.

A conjunção destas duas conjeturas, com a massificação de *headsets* de realidade virtual e das tecnologias de visualização que com eles prosperam, abre um espaço atrativo para a introdução de serviços de visualização baseados em realidade virtual com foco na progressão da pesquisa científica.

O objetivo desta dissertação é explorar e avaliar a recetividade das entidades competentes à introdução de realidade virtual como uma ferramenta de visualização científica no campo da biofísica computacional.

A realidade virtual é uma *interface* de simulação de ambientes tridimensionais, gerados por computador, onde o utilizador pode interagir de forma aparentemente natural com a cena. Uma das formas de implementação desta *interface*, que foi estudada no presente documento, corresponde à utilização de um *headset* no utilizador, auxiliado por dois controladores de mão, numa aplicação de visualização interativa.

A estratégia utilizada para quantificar a avaliação proposta passou pelo desenvolvimento de um protótipo de uma aplicação (baseada na *web*) com visualização de dados científicos através de realidade virtual e submissão deste a um conjunto de testes práticos, qualitativos e quantitativos, com respetivo *feedback* por parte dos intervenientes. Paralelamente foi elaborado um estudo, segundo o modelo de aceitação de tecnologias.

A solução proposta procurará constituir uma base de trabalho sólida e suficientemente genérica para ser aplicada em contextos onde a visualização de informação científica através de realidade virtual possa ser bem-sucedida.

Foram efetuadas 3 avaliações (avaliação de aceitação, avaliação prática ao protótipo e avaliação teórica às interações do protótipo) que, embora condicionadas por um número de participantes reduzido, permitiram a obtenção de um feedback positivo sobre o trabalho desenvolvido.

Palavras-Chave: realidade virtual, biofísica computacional, modelção e simulação, software científico, visualização de dados,

Abstract

Scientific visualization, an interdisciplinary concept between science and computer graphics, aims to allow a visual conceptualization in the scientific research process. The ability of scientists to visualize the result of computations and simulations is essential for the acquisition of new knowledge and, therefore, for the evolution of the surrounding scientific field.

The scientific method is, by its nature, an interactive and iterative process which requires that tools to aid graphic visualization are the least possible limiting factors with regard to interaction with the environment.

The combination of these two conjectures, with the massification of virtual reality headsets and the visualization technologies that thrive with them, opens an attractive space for the introduction of visualization services based on virtual reality with a focus on the progress of scientific research.

The objective of this dissertation is to explore and evaluate the receptivity of competent entities to the introduction of virtual reality as a scientific visualization tool in the field of computational biophysics.

Virtual reality is a simulation interface of three-dimensional environments, generated by computer, where the user can interact in an apparently natural way with the scene. One of the ways of implementing this interface, and which was studied in this document, corresponds to the use of a headset by the user, aided by two hand controllers, in an interactive visualization application.

The strategy used to quantify the proposed evaluation involved the development of a prototype of an application (web based) with visualization of scientific data through virtual reality and submission of it to a set of practical, qualitative and quantitative tests, with respective feedback from the participants. At the same time, a study was prepared, according to the technology acceptance model.

The proposed solution seeks to constitute a solid and sufficiently generic work base to be applied in contexts where the visualization of scientific information through virtual reality can be successful.

We realized three evaluations (acceptance evaluation, practical evaluation of the prototype and theoretical evaluation of the prototype interactions) which, although conditioned by a reduced number of participants, allowed to obtain positive feedback the work developed.

Keywords: virtual reality, computational biophysics, modelling and simulation, scientific software, data visualization

Acknowledgements

To my advisor, Prof. Augusto Sousa, and to Prof. António Coelho for their support and encouragement during the dissertation, for their help in reviewing the work registered here and for the opportunity to study an area as exciting as this.

To MetaCell's staff, for their constant willingness to answer the various questions that were asked and for their precious help at different moments of the realization of this dissertation, especially during the prototype development.

To the friends I created in college, and with whom I shared great moments over the past 5 years.

To my family, for their understanding and support and, especially, to my brother and girlfriend for their help in reviewing this document.

Afonso Pinto

"L'essentiel est invisible pour les yeux."

Antoine de Saint-Exupéry

Contents

List of Figures

List of Tables

Abbreviations

2D	Two Dimensions
3D	Three Dimensions
CAVE	Cave Automatic Virtual Environment
DCM	Direction Cosine Matrix
EFA	Exploratory Factor Analysis
fps	frames per second
HMD	Head Mounted Display
HTTP	Hyper Text Transfer Protocol
IMU	Inertial Measurement Unit
NFI	Normed Fit Index
PI	Place Illusion
PLS	Partial Least Squares
Psi	Plausibility Illusion
VR	Virtual Reality
SRMR	Standardized Root Mean Square Residual
SSL	Secure Sockets Layer
TAM	Technology acceptance model
TLS	Transport Layer Security

Chapter 1

Introduction

This dissertation involves both virtual reality and computational biophysics areas, with particular focus in visualization tools for scientific data analysis. This chapter will provide an overview on the motivation, context and objectives of the dissertation.

1.1 Context and Motivation

Over the past two decades, computational science has gained more and more relevance in the research academy, industry and laboratories to advance discovery. Its ability to provide scientists and engineers with powerful, reproducible and shareable models/experiments has become a prevalent mean of discovery and innovation in essentially all areas of science, allowing them to face new, more exciting and more complex challenges. However, as the scientific problems under investigation becomes more complex, the amount of data that scientist have to deal with grows. The complexity and quantity of data generated by today's computational scientific tools can make it impossible to process information completely numerically.

Visualization has become a necessity for understanding both large and complex amounts of data generated by the current scientific problems. If it is true that a number of visualization techniques can automatically extract features of interest, it is also true that the majority of the techniques require human intervention in order to understand and interpret the complex structures in study. Adding interaction (either by allowing the exploration of a three-dimensional scene or by allowing the tuning of some visualization parameters) to the receipt, greatly increases a researcher's perception and understanding [103].

> "Interactively exploring a three-dimensional scene can greatly improve a researcher's understanding of the three-dimensional relationships in an environment through motion parallax." [48]

With scenarios that require this kind of visual interactivity, we decided to explore an emerging technology, characterized by the immersiveness that provides, which recently saw huge improvements in the hardware that requires and whose prices are at reasonably affordable levels, to stimulate the formulation of hypotheses, facilitate the discovery of causality and assess available evidences. The aforementioned technology is VR.

The dissertation was developed in partnership with a neuroscience software company called MetaCell[1]. MetaCell is the world leader in software for neuroscience. It partners with academic institutions and pharma organizations, providing software for simulation and data analysis and helping them make the most of their neuroscience data and models.

The main goal of this project was to assert if the use of VRin visualization and exploration of scientific data in the fields explored by the company, namely computational biophysics, taking advantage of a more immersive environment, would be a featured accepted by the relevant entities and if it would speed up the verification of theories and new scientific discoveries.

1.2 Problem Statement

This dissertation addresses the problem of trying to measure the relation between the level of interactivity and immersiveness experienced by scientific researchers when using visualization tools and the quality and quantity of their results. More specifically, we want to understand the performance impact of using VR(using head mounted displays) when compared with the current use of standard 3D desktop visualization.

1.3 Objectives

The main objective of this dissertation is the study of the benefits of introducing VR in computational biophysics visualization tools.

As an illustration of this technology's usefulness, a prototype of a very simplistic web-based application with an immersive VR experience for neuroscience data visualization was implemented.

Within the scope of this main objective, specific objectives can also be identified:

- Preparation of a state-of-the-art technology report;
- Investigation of visualization techniques in VRas well as problems existing in each of them;
- Creation of a prototype of a web-based application with an immersive VR experience that allows simple interactions with the data visualized (selection, rotation, scaling and position changes);
- Implementation of a set of experiences in view of the problem under study;
- Prototype testing and evaluation.

[1] https://www.metacell.us/

1.4 Contributions

The main contributions of this dissertation include:

- The development of a simplistic visualization application for scientific data in the domain of computational biophysics, based on immersive VR technologies. This application becomes the foundation of this thesis research, which is used as a tool for supporting performance analysis and gather feedback.
- A set of benchmark methodologies to quantify all performance parameters of the prototype application. This prototype application and its associated micro-benchmarks make up as a tool for performance evaluation and analysis.
- An analysis based on the Technology Acceptance Model (TAM), in order to study the acceptance of the introduction of VR in visualization tools for scientific data.

1.5 Related Work

In the area of scientific visualization using VR, Bryson's paper of 1996 [14] stands out giving a very comprehensive first approach on this issue, with focus on the performance requirements of the given system. A practical use of VR for scientific visualization is covered in [102] although this one is done in the context of big data management. Still regarding the use of VR in scientific visualization, but also framed in the topic of recommended strategies for visualization and interactions in VR [52]. Several surveys were also consulted, which proved to be very useful in identifying the most common problems of current 3D visualization systems, some of them being [19] [10]. Moving now to the domain of quantifying the benefits of changing a visualization system from 3D to virtual reality, [62], despite being an article whose focus is augmented reality, shares most of the concerns to be had when implementing an evaluation method of the same type as the one on the present thesis. The paper [59], although focused on the use of VRhardware in general, also addresses the acceptance of this technology according to an extended technology acceptance model.

1.6 Document Structure

This document is divided into six chapters: Introduction, Literature Review, Methodology, Implementation, Evaluation and Discussion and Conclusions and Future Work. This chapter seeks to provide an overview of the topic. Contains a presentation of the context and problem in a synthetic way, as well as the goals of the dissertation. In the Literature Review, the state of the art is described for each of the areas related to the dissertation theme, and several alternative solutions are presented for each of the problems encountered. Then, in the Methodology chapter, is approached, in a more theoretical way, the solution found and the work developed during the dissertation. In this section, the theoretical foundations of the approached methods are presented and, whenever necessary, the empirical examples that served as the basis for the decisions made. In the next chapter, Implementation, the prototype implemented during the dissertation is described and the most important technical issues that characterize the proposed system are identified. The following chapter, Evaluation and Discussion contains the description and discussion of the evaluations performed. Finally, in the last chapter, possible concrete applications of this solution are mentioned and pointed out some basic directions for the work that may be developed in the future.

1.7 Summary

This chapter provides an overview of the topic. Contains a synthetic presentation of the context and problem, as well as the goals of the dissertation. It also states the contributions of this thesis and related work.

The next chapter will address the state of the art of the relevant fields of this dissertation: Computational Biophysics, Visualization, 3D Visualization and VR.

Chapter 2

Literature Review

Throughout this chapter, the state of the art of each one of the areas related with this dissertation will be addressed. The first section is reserved for the state o computational biophysics (section 2.1). In the second, we will introduce the concept of visualization (section 2.2). The theme of the third section is 3D visualization (section 2.3), where it will be discussed current strategies to develop, as well as identifying some of their common problems. In the forth and last chapter, the concepts of VR(section 2.4) will be reviewed, as well as its evolution over the years. This topic will be the most relevant in the context of the chapter, given that this is the main component of the evaluation to be performed in this dissertation and, at the same time, the area that raises the most scientific interest given its recent emergence. Throughout this analysis, several alternatives will be presented and discussed, giving privilege to those who will, at the outset, be more suitable for an end use of according to thesis requirements.

2.1 Computational Biophysics

Computational biophysics is a field at the intersection between computer science, physics, chemistry and biology. It is described by Ilan Samish et al. as:

> “Computational biophysics is a hypothesis-driven physics-based treatment of biological systems.” [82]

Researchers in this area try to understand the influencing factors, the functions and the interplay between different parts of complex biological systems. The field has become increasingly popular during the last few decades. One factor that has greatly contributed to this growth is the increased availability of computer power. As biology advanced into this century, new levels of quantitative understanding of biological systems were required; if the quality of the hardware had not kept up with those requirements, some of the exciting developments of today would still be beyond imagining.

No single approach can fully characterize the research that falls into this general area and therefore, in this section (and in this dissertation in general), we will use computational neuroscience (a sub branch of computational biophysics) as a concrete example of computational biophysics.

2.1.1 Computational Neuroscience

As part of the larger field of computational biophysics, computational neuroscience (CNS) tries to understand how does the brain (and the complete nervous system) generate behaviors using computational approaches. The organization for computational neurosciences defines the term "computational neuroscience" as follows:

> “Computational neuroscience is an interdisciplinary field for development, simulation, and analysis of multi-scale models and theories of neural function from the level of molecules, through cells and networks, up to cognition and behavior. ” [83]

This definition refers a key components of modern research in neuroscience, ’models’. The next subsection will explore this concept in detail.

2.1.1.1 Computational Modelling

Computational models have become an important tool in the study of the nervous system and are commonly used in the simulation of specific aspects of physiology and pathology [2]. The main goal for using computational modeling is to understand the behavior of complex systems using mathematical analysis and computer simulations [66].

A model can be seen as a simplification of the system where we want to test some hypothesis. Thomas Trappenberg describes it as:

> “A model is abstraction of a real-word system to demonstrate particular features of, or investigate specific questions about, the system. Or, in more scientific terms, a model is a quantification of a hypothesis to investigate the hypothesis.” [99]

Models are intended to simplify/abstract complex systems, thereby identifying which details of the whole are essential to explain a certain phenomena. Additionally, by being formalized mathematically, the assumptions of the model are explicit, unambiguous and logically consistent, making them easily reproducible.

Further subsections will provide answers to typical questions when deciding to build a computational model: What level to model? How detailed the model should be? How to model? How to use the model?

Level of Analysis

No single neural model can be expected to span all the levels of the nervous system (Fig. 2.1) and an essential feature at one level of organization may be an insignificant detail at another. The nature of the scientific question that drives the modelling work is what determines the level at what the model is to be constructed [89].

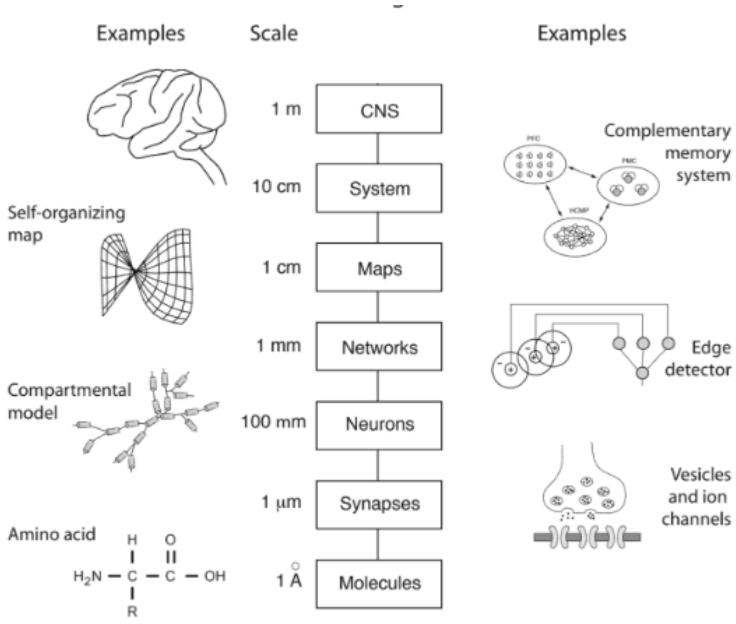

Figure 2.1: Multiscale levels of investigation for studying the brain. Adapted from [21]

Level of Detail

The level of detail of a model closely relates with how realist we want the model to be. This decision should not be taken lightly as the choice for a more detailed model comes at the cost of higher model complexity and higher computationally power required. Present constraints limit simulations to tiny nervous systems or small components of more complex systems [84].

- **Realistic Models** - This strategy consists of a very large scale simulation that tries to incorporate as much of the cellular detail as is available. An example of a realistic model at the level of a single neuron is the Hodgkin-Huxley model [41].
- **Simplified Models** - This approach consists of models that reproduce the essential properties of physical systems. On one hand, since we do not yet know all the cellular details, there might be important features that are being inadvertently left out. On the other hand, simplifying models of the brain can provide a conceptual framework for isolating the basic computational problems and understanding the computational constraints that govern the design of the nervous system. The scaling of this strategy is an important aspect for practical applications.

Methods

Another factor to consider when deciding which level of detail to adopt is the set of methods that are going to be used for modelling, since they can have a substantial impact in the complexity of the model. In this section we will address the two principal categories of methods that account for most of the research field. These methods are: 1) Single neuron models and 2) Network models.

Single Neuron Models Neurons are specialized cells with a high level of polarization defined by the presence of three major compartments: the dendrites, a cell body (or soma) and an axon (Fig. 2.2). Dendrites are specialized to receive electrochemical signals, which are then processed and transferred through the cell body and along the axon to be transmitted to the target cell/s. [28].

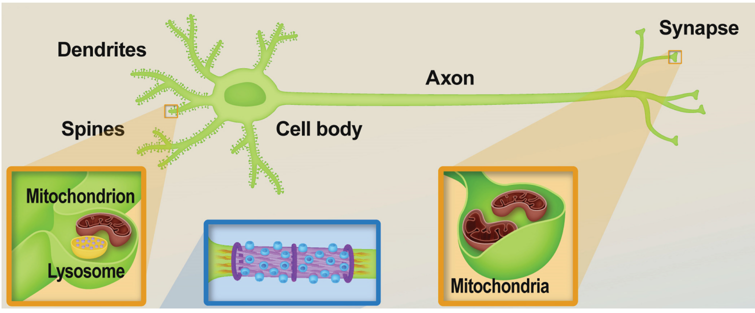

Figure 2.2: Neurons are polarized cells with three main compartments: dendrites, cell body (or soma), and axon

Many models have been proposed to modulate the behavior of single neurons, ones extremely complex and others more simplistic.

- Compartmental Models - Are the most detailed class of neuron models. In this modulation technique the cell is modeled as a set of compartments and each dendritic compartment is modeled as an electric circuit (Fig. 2.3) with the equation :

$$C\frac{dV}{dt} = \sum_j g_j(y)\,[E_j - V] + \frac{V_r - V}{R} \tag{2.1}$$

  where V is the membrane potential, Vr is the resting membrane potential, R is the passive membrane resistance, C is its capacitance, Ej is the reversal potential of synapse j on the path, and gj(t) is the time-varying conductance of synapse j and dV/dt represents the rate of change for V with respect to time [66].

  Typically, simulations with compartmental models focus on single cells, or a small number of cells due to performance issues.

- Integrate-and-Fire Models - These models are a simplified version of the compartmental models. This simplification is divided in 3 crucial changes:

  1. The entire neuron is modeled by a single compartment.
  2. Synapses are model as parameters and not as ion channels.

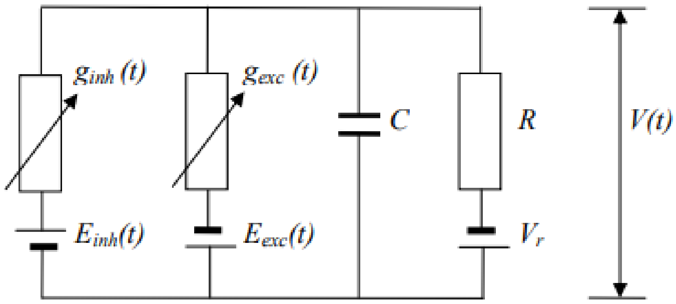

Figure 2.3: Equivalent circuit model of a dendritic compartment.

3. The action potential generation is modeled by:

$$C\frac{dV_i}{dt} = \frac{V_i}{R} + \sum_j w_{ij} \sum_k \partial(t - t_{jk}) \tag{2.2}$$

   where wij is the weight (efficacy) of the jth synapse, tjk is the time at which the kth spike arrives at synapse j, and d(t) is the Dirac delta function representing the spike [66].

This modulation technique have been used mostly in network-level modeling of the nervous system.

- Rate Models - These models are a simplification of the integrate-and-fire models. They replace the $\partial(t - t_{jk})$ of eq. 2.2 with a continuous variable, typically

$$f_j(t) = [1 + exp(-\lambda V_j]^{-1} \tag{2.3}$$

  to represent the spiking rate.

  Rate models are the simpler models than Integrate-and-Fire and are widely used in neural network models.

- Threshold Models - The simplest class of neuron models which simply views neurons as all-or-none devices. This is due to the replacement of eq. 2.3 with a binary threshold. A cell that is sufficiently excited has a firing rate of 1 (active), while an insufficiently excited cell has a rate of 0 (inactive).

  This models, although very simplistic, allow the simulation and analysis of very large-scale networks, providing very valuable intuitive results regarding the collective behavior of such systems [42].

Network Models The most valuable information in the nervous system occurs not in the individual but at the collective level, so rather than focusing in single neuron models, the primary interest of computational modelling goes on the behavior of networks. Two neural regions can be very different from each other, but there are some canonical types of network architectures that have been identified as being of general interest no matter the anatomy:

- Feed-Forward Network - This architecture is characterized by signals flowing unidirectionally from one set of neurons to another (without any feedback). Few are the parts of the brain that do not have feedback actions, but for the ones who don't and the ones where those effects can be ignored (directed projections from the principal neurons of one area to another) feed-forward architectures are a good way to go. A well-known instance of a feed-forward network is Hubel and Wiesel's model of feature detectors in simple cells and complex cells of the primary visual cortex [43] [44].

- Recurrent Networks - In this architecture the outputs of cells in a layer feed back to the same cell population or to upstream populations. This strategy is gaining more popularity as new discoveries in the nervous system lead to a substantial relevance of recurrent systems in the brain (e.g. the thalamocortical loop [81], CA3 cell and relations with associative memory [60]). Figure 2.4 shows an example of a network containing two groups of excitatory neurons (increase the likelihood that the neuron will spike) in a cell layer with both recurrent self-excitation and recurrent mutual inhibition.

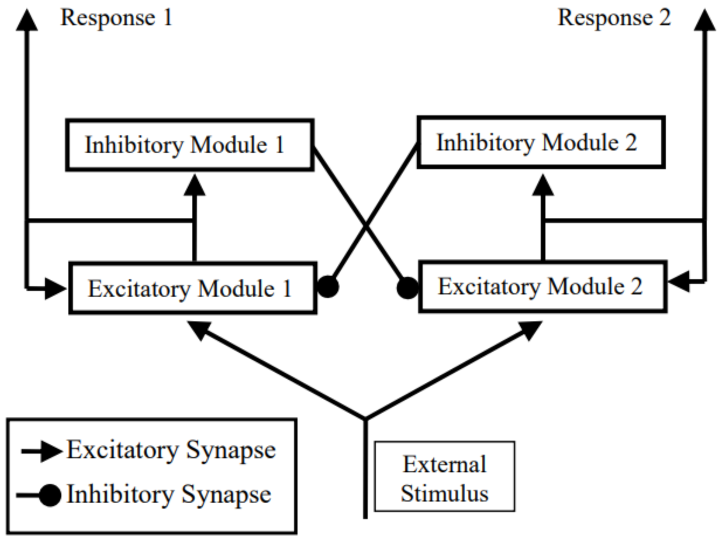

Figure 2.4: A competitive recurrent network.

2.1.1.2 Computer Simulations

After building the model, one can now use it to test or prove new theories by iteratively change its parameters and evaluate the resulting changes. Traditionally this was done by performing applied mathematics and theoretical physics with a pen and a paper. However, with the appearing of computational modeling, we can now use 'experimental mathematics'. This process consists in running virtual experiments to try understand the behaviors of the model. Compared to the traditionally method it brings the following advantages [57]:

- While traditional methods provide a clean reality, simulations provide an alternative reality, easier to manipulate and access.
- The scope of the model complexity (in terms of calculus) that can be target is incomparably superior.
- Running computer simulations allows one to test specific questions about causality that could only be guessed in a paper-and-pencil modeling.

2.1.1.3 Computational Neuroscience Tools

So far we have seen that being able to run computer simulations is currently a very important feature in computational neuroscience. Another crucial factor, which will be described in detail later on this chapter, refers to visualization. Multiple studies (e.g [46] [51] [104]) suggest that the visualization of 3D models can facilitate the learning of complex conceptual relationships. Therefore, auxiliary neuroscientific tools of today should be able to provide both simulation and visualization features. However, it is often far from trivial to use both of this features together [16].

In one hand, for simulation, the principal set of tools we have available contain: NEURON [40], which according to [97] is better suited for detailed models; BRIAN [37], which provides the most concise language for both large and small networks, NEST [75] which mostly favors large network models and GENESIS [7]

On the other hand, for visualization, various 3D visualisation tools have been developed such as BioLayout Express3D [32], Arena3D [73], Amira 3D [88], V3D [74], the Allen Brain Atlas [56] or Cytoscape [85]. All these tools are very complete, but require local installation and are usually complex to operate for non-expert users.

Typically, in order to solve this fragmentation problem, developers use general purpose programming languages such as Python, to generate the tool-chains necessary for their needs. However, this solution contributes to the technological division of the field since the tool-chains developed are usually case specific and in some cases inaccessible to many researchers. Such technological barriers have had a remarkable effect in the neuroscience field as a whole, resulting in computational models that are poorly validated and in unexplored model-generated hypotheses [18].

Tools like, Geppetto [18] and Visimpl [34], try to combat that problem by providing an aggregated platform to both visualizing neuroscience models and managing simulations.

2.1.1.4 Future directions

In the past decades, digital computers have been increasing in computing power and data storage which transformed how computational neuroscientific experiments are performed, how scientific data are analyzed and the development of brain models and theories. Looking into the future, the field of computational neuroscience promises new opportunities and challenges. The

Brain Initiative 2.0 [67] defines the following priorities for the development of computational neuroscience in the next 5 years (ending in 2025):

1. Discovering Diversity: Identify, determine their roles in health and disease and provide experimental access to the different brain cell types.
2. Maps at Multiple Scales: Generate multi-scale circuit diagrams from synapses to the whole brain.
3. The Brain in Action: Improve methods to monitor large-scale neural activity and try to produce a dynamic picture of the functioning brain.
4. Demonstrating Causality: Improve simulation tools so that precise changes in the neural circuit dynamics can lead to links between brain activity and behavior.
5. Identifying Fundamental Principles: Analyse data and try to produce conceptual basis of understanding for the biological act of mental processes.
6. Advance Human Neuroscience: Develop innovative technologies to try understand the human brain and treat its neurological disorders.
7. From the BRAIN Initiative to the Brain: Aggregate the features/tools resultant from goals #1-6 into a centralized tool to facilitate the discovery of how dynamic patterns of neural activity are transformed into cognition, emotion, perception, and action in health and disease.

Despite all the scientific and technological progress so far, computational neuroscience is still in its early stages and it is expected that in the near future we will see improvements on the modelling studies and their projections of the highly complex biological reality of the human brain.

2.2 Concept of Visualization

The term visualization can be used to name different means, in this thesis, we focused on visualization in computing, which may be technically referred as computer-supported data visualization. In this context, G. Scott Owen defines visualization as follow:

> “Visualization is essentially a mapping process from computer representations to perceptual representations, choosing encoding techniques to maximize human understanding and communication.” [70]

McCormick goes further and adds the scientific discovery factor to the definition itself:

> “Visualization is a method of computing. It transforms the symbolic into the geometric, ... Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights.” [25]

In the above definition, there is a mention to the term "insight". This is a non-trivial concept which is defined by multiple dictionaries as "accurate and deep intuitive understanding". If it is true that this is in fact the objective of those who create and use visualization tools, it is also true that it is a rather complicated notion to objectively measure and evaluate.

Given this perhaps too much vagueness, in the remainder of this dissertation document, when we refer to visualization we will be thinking about the following definition:

> “Visualization is a study of transformation from data to visual representations in order to facilitate effective and efficient cognitive processes in performing tasks involving data. The fundamental measure for effectiveness is correctness and that for efficiency is the time required for accomplishing a task.” [20]

2.2.1 Sub-fields of Visualization

Currently the global concept of visualization is typically used into two sub-fields:

Scientific Visualization: displays spatial data associated with scientific processes such as the bonding of molecules in computational chemistry.

Information Visualization: develops visual metaphors for non-inherently spatial data such as the exploration of text-based document databases

The dividing line between these two fields is whether the spatialization is given (scientific) or chosen (information). However, more recently, there seems to be an effort on the part of some members of the visualization community to shrink that line and bring these two fields together. This current of thought is supported by the unprecedented amount of information available from large-scale simulations, experiments, and data collection to scientists today, making the traditional scientific visualization field not well suited for the challenges of today [79]. The merge with information visualization might come in handy, as the latter is described by Alexandre Valle de Carvalho as:

> “Information visualization focuses on compactness, meaning the ability to graphically compact large amounts of information in such a manner that allows the observer to more effectively discover, make decisions or provide explanations about patterns individual or groups of information items.” [24]

2.2.2 Visualization Advantages

In order to show some of the visualization advantages, let us consider, as example the task of analyse a stream of numbers compared with graphically visualise the same data:

- Faster Observations - Viewing a graph is much faster than viewing a stream of numbers, since the first facilitates the pre-attentive processing, allowing information to be obtained from the environment unconsciously. [91].

- Stimulating Hypotheses and Other Thoughts - There are some studies suggesting that when visualization is appropriately designed and allows clear and natural interactions, it can help understand the underlying data, find relationships and stimulate hypotheses more effectively [20] [24].

- Evaluating Hypothesis - The scientific method (Fig.2.5) consists of 6 steps. The 5th refers to hypothesis testing and it is critical in this process. Whenever applicable and doable, one should utilise verified testing methods, such as Bayesian hypothesis testing [6]. However, those methods require non-trivial amount of time and work to process. In practice, visualization is often used as an intuitive form of hypothesis validation. Typically this process consists in comparing the results of a simulation (hypothesis) with some ground truth data [20].

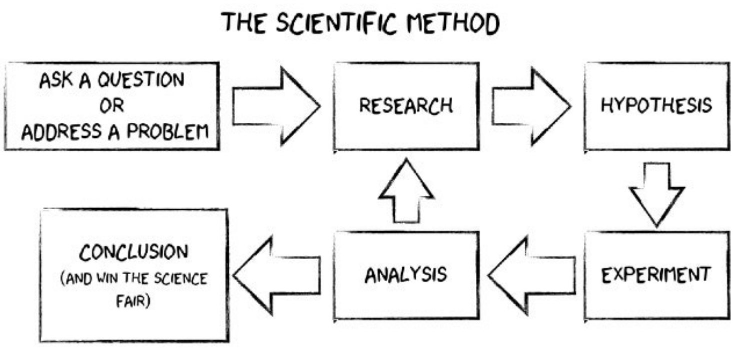

Figure 2.5: Diagram of the scientific method

2.2.3 The Visualization Pipeline

We can describe the step-wise process of creating visual representations of scientific data the visualization pipeline [52]:

- **Simulation**: results of numerical simulations are generally the input of the visualization pipeline.

- **Data Selection & Filtering**: in this pre-processing step the raw data is putted in a specific format by application of for example, smoothing filters, interpolation of missing values or corrections of erroneous measurements.

- **Visualization Mapping**: in this stage the processed data from the previous step is mapped into the visualization space, i.e. transformed into graphical primitives. According to Mackinlay [58], this is the most critical step to achieve expressiveness (quality of the graphical language that expresses the desired information) and effectiveness (quality of the graphical language that exploits the capabilities of the output medium and the human visual system)

- **Rendering**: At last, the graphical primitives are rendered as images, which are then displayed on the screen(s).

This sequence of steps (Fig. 2.6) shall be seen as an interactive and iterative process, allowing the scientist to explore relevant features or patterns in the data.

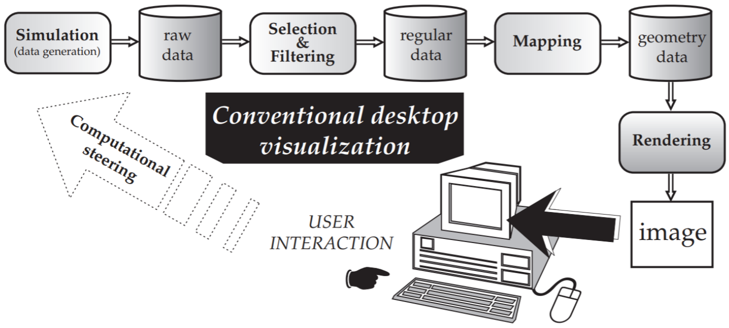

Figure 2.6: Visualization pipeline on desktop workstation

By allowing the fine-tuning of the visualization model until the desired result is obtained, we facilitate an exploration of the type 'what if' analysis, which with enough computational power can lead to an enhancement in productivity through the reduction in time between optimization of visualization control parameters and viewing the results [48].

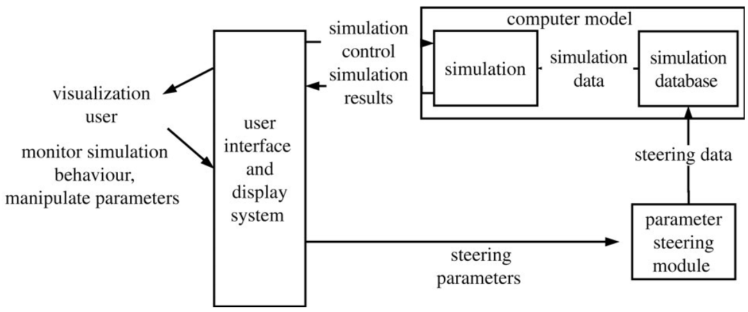

Figure 2.7: Diagram of a 'what if' type of analysis system

2.3 3D Visualization

So far we have defined what visualization is and what it can be used for. It's time now to describe how visualization tools of today typically perform the visualization of scientific data. Traditional tools used to rely on 2D images displayed on a monitor; this forced the viewer to use auxiliary visual cues in order to judge the position and depth of the data. Due to higher demands in terms of data quantity and complexity and improved visualization technologies, the standard is now try to create visualizations that are closer to real-world with 3D visualization. Both 2D and 3D visualization strategies have value for tasks involving 3D spatial data. Springmeyer et al. [87] observed that 2D views are often used to establish precise relationships, whereas 3D views are typically used to gain a qualitative understanding of the data and present that understanding to others. In [93] it is referred that 3D visualization appears to greatly enhance the ease and efficiency of basic data interpretation, produces no more eye strain or headaches than 2D visualization and is overwhelmingly preferred by the viewers.

In the next sections we will provide a basic introduction to 3D Graphics core concepts and terminology as well as an introduction to the standard 3D graphics library for the web and lastly the common issues associated with 3D visualization systems.

2.3.1 3D Graphics

In his book, Tony Parisi uses the following concept of 3D graphics:

> "3D computer graphics, or three-dimensional computer graphics (in contrast to 2D computer graphics), are graphics that use a three-dimensional representation of geometric data (often Cartesian) that is stored in the computer for the purposes of performing calculations and rendering 2D images. The resulting images may be stored for viewing later (possibly as an animation) or displayed in real time." [72]

This notion can be break down into 3 components: [72]:

- The data is represented in a 3D coordinate system (Fig.2.8);
- It is ultimately drawn (rendered) as a 2D image (for example in a computer monitor);
- When dynamically updated, the image is rendered without a perceivable delay.

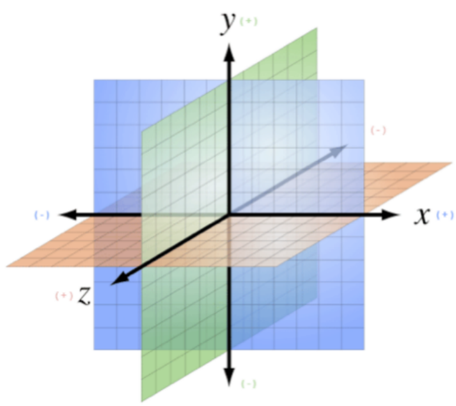

Figure 2.8: A 3D coordinate system

The most common way to draw 3D graphics is by the use of meshes (Fig.2.9). A mesh is an object composed for one or more polygonal shapes, constructed out of edges and vertices (x,y,z) defining the coordinate positions in the 3D space.

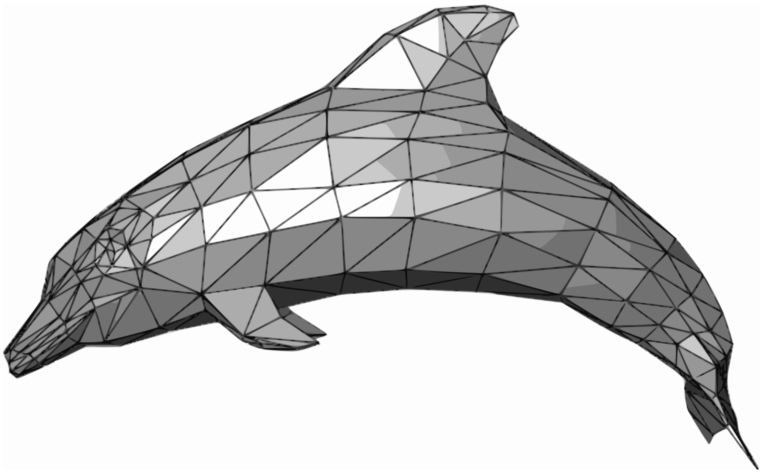

Figure 2.9: A 3D mesh

The way that an object looks and the amount of light it reflects or emits can be emulated by the use of different textures and materials respectively. Changing the position of 3D objects is typically made through the use of transforms, which are operations that let you scale, rotate and translate a rendered mesh.

3D visualization systems typically use a camera to define where the user is positioned and oriented in the scene as well as its field of view. Another crucial aspect of the camera is to deliver the final rendered image of the 3D scene into the 2D viewport defined by the window or canvas. This two distinct tasks are normally matched by the use of two distinct matrix to represent the camera, the position/orientation matrix and the projection matrix, respectively.

The final image of a mesh can finally be rendered with the use of a shader (also known as programmable shader), which is programmed to interpret the high-level structures defined (materials, lights, transforms and cameras) that the graphics hardware can't process.

2.3.2 WebGL

Traditionally, 3D graphics were restricted to high-performance computers or dedicated game consoles, and required complex programming. However, nowadays hardware is shipped in every computer and mobile device and besides that, we also have the software required to render 3D accessible for free within web browsers. This is possible since the adoption of WebGL (Web Graphics Library) by current web browsers [33].

WebGL is a JavaScript API, based on OpenGL Embedded System (ES) 2.0 [55], that can be used to render high-performance interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins [107].

The ease of use and accessibility of WebGL is what differentiates this technology from others. It produces an advance in usability with respect to the obsolete need to locally install desktop applications. WebGL is a multi-platform technology and because it is/produces web applications it becomes more flexible in terms of fetching, handling and sharing resources [61], which are key points for scientific research.

The evolution of web standards and protocols[1] preceded the appearance of WebGL 1.0. In 2011, this technology was developed and integrated into all of the Web standards of the browsers. In 2017 the second generation of WebGL, called WebGL 2.0, was released with support for real-time rendering and VR (Figure 2.10) [107].

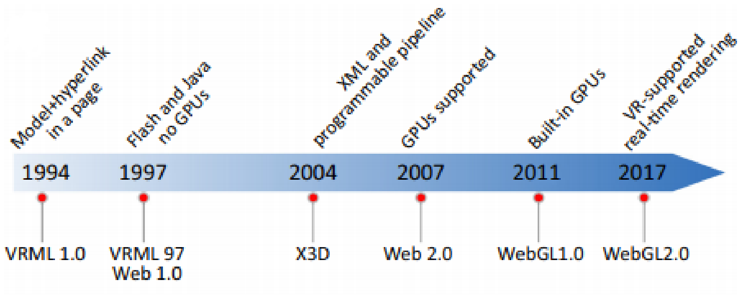

Figure 2.10: The History of Web3D

With the introduction of WebGL, the GLSL ES shader language was added to the technological stack of browsers. Figure 2.11 compares the traditional web page architecture (left side) with the web page architecture using WebGL (right side) [61].

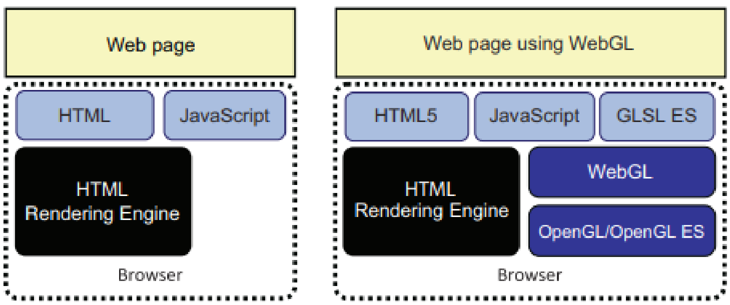

Figure 2.11: Traditional web browser workflow (left side) and web browser workflow using WebGL (right side)

2.3.3 Problems

Performing interactive 3D visualization on regular 2D screens is prone to problems. One of the typical issues pointed against 3D visualization is the difficulty for users to navigate in 3D spaces using 2D input devices (mouse, tablet, trackball etc.); correlating 2D movements in the real world to movement of objects in the virtual world can be challenging, especially if the planes of movement are not similar [39].

Other complain, referring mouse-based environments, is that the user may desire to do various click-and-drag operations including object selection, scene pan, scene zoom and scene rotation;

[1] As an example, VRML (Virtual Reality Markup Language) is a standard file format for representing 3D interactive vector graphics that evolved to X3D [12] with the development of XML technology.

The management of all those interactions - navigation technique - adds extra complexity to the model interpretation [10].

Another common issue of 3D visualization tools refers to selection and manipulation of objects. Actions to select and navigate, usually overlap, (the common 2D click and drag interaction to create a selection bounding box overlaps the common drag action used for navigation), making the cognitive effort needed by the user increase due to the necessity of managing both operations [10].

Occlusion may also be a problem, as it can distort the user's perception of the scene, mainly when the information space is dense [96].

3D perspective perception is a complaint as well. Figure 2.12 shows an example of a misleading 3D chart. In 3D, objects closer to the user appear to be larger than those in the back.

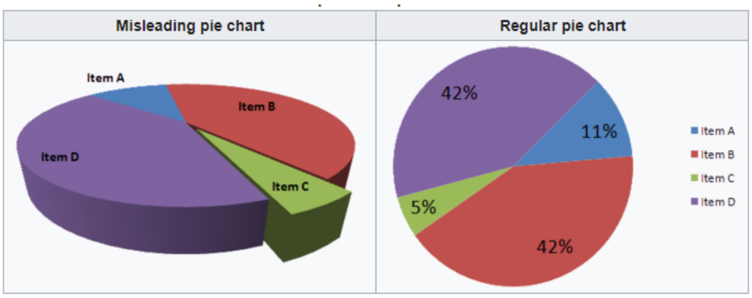

Figure 2.12: Comparison of pie charts - Item C appears to be at least as large as Item A, whereas in actuality, it is less than half as large

2.4 Virtual Reality

The original term VR can be described as a computer generated 3D environment in which the user is being enveloped by, included in, and interacting with [52].

In [13], Bryson defines it as:

> "VR is the use of computer technology to create the effect of an interactive 3D world in which the objects have a sense of spatial presence." [13]

This definition, although correct, fails in distinguish VR from 'other realities'. To do so, we will resort to Paul Milgram et al. work on the "reality-virtuality continuum" [64] (Fig. 2.13) which defines the terminology and distinguish the different disciplines of the field.

On the right extremity we have VR environment which correspond to the completely synthetic world, which may or may not mimic follow the real-world environment rules. In contrast, on the left side extremity of the spectrum we have the real-world environment. Augmented Reality is placed between real environment and virtual environment, closer to the former. Both of Augmented Reality and VR share interactivity and three dimensional images, between other factors. Yet, the differences between them are quite evident:

1. The level of immersion: Augmented Reality never lets a user lose a sense of presence in the real world as it just supplements some part of reality with virtual factors. In opposition, VR completely consists of computer-generated factors, which makes a user totally immersed in it.

2. Users movement: Augmented Reality typically requires portability of the system, while VR limits an user's physical movement to specified region.

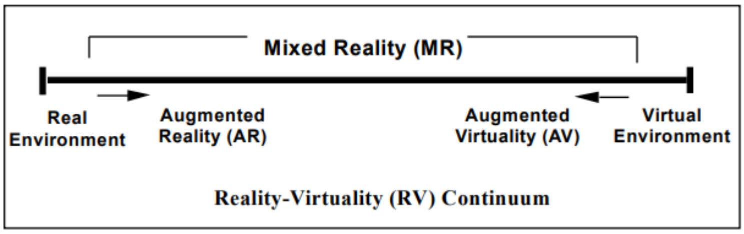

Figure 2.13: Milgram's Reality-Virtuality Continuum

2.4.1 History

The present section will explore some of the historical milestones (see Fig.2.14) that have led to the advent of VR technologies.

The first attempts to immerse a person in a different place date back to the mid-19th century (1838) with the Charles Wheatstone stereoscope, which demonstrated that the brain processes the different two-dimensional images of each eye into a single three-dimensional object.

Another important contribution to the VR world happened in the mid-20th century (1950): the development of "the sensorama", an arcade-style theatre cabinet that would stimulate all the senses, by Morton Heilig. A decade later (1960), he produced the first monitor mounted on a VR head, the "telesphere mask". The headset provided stereoscopic 3D and wide vision with stereo sound, but without motion tracking.

Ivan Sutherland's "ultimate display" was conceptualized 5 years later (1965) and is seen as the core blueprint of the concepts that today encompass Virtual Reality. It describes both concepts of a realistic virtual world, seen through an HMD, with enhanced 3D sound and tactile feedback and the ability to interact with it realistically and in real-time.

Between 1977 and 1982, the first finger tracking gloves for VR, called "Sayre", were invented by Daniel Sandin and Thomas DeFanti.

In 1989, NASA completed the creation of Project VIEW, a VR simulation used to train astronauts.

1991 is the year the first reliable HMD, priced less than 10,000 \$, VR-2 Flight Helmet from Virtual Research Systems, was launched. In the same year, the CyberEdge Journal, the first commercial newsletter for the VR community, was published.

The end of the millennial and the beginning of century 21st, is marked by a decrease of commercial interest in VR technology reflected by the failures of SEGA's VR glasses and Nitendo's

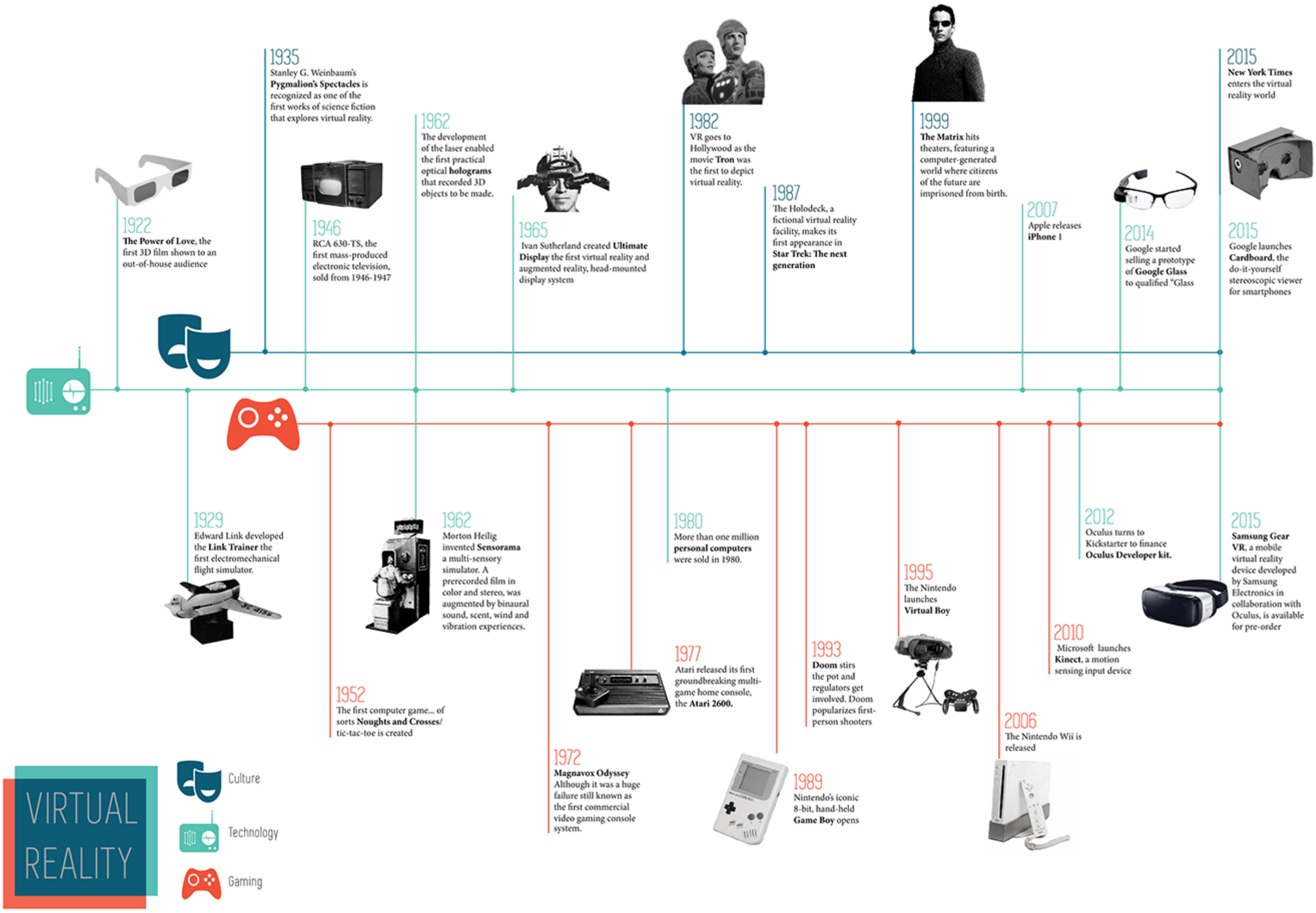

Figure 2.14: The evolution of VR

Virtual boy, due to a mix of high cost, quality below expectations and reports of motion sickness and nausea.

It was not until recently that high-quality VR devices started to reach the consumer market at affordable prices; this change is seen as the main responsible of the recent emergence of VR (Figure 2.15) .

2.4.2 Core Concepts

In order to achieve the psycho-physical experience of being present in a virtual environment, we need to integrate VR Hardware with a responsive computer-generated 3D environment. In 2003, Michael Zyda and Tom DeFanti [108] summarized four key concepts that should allow VR to provide that:

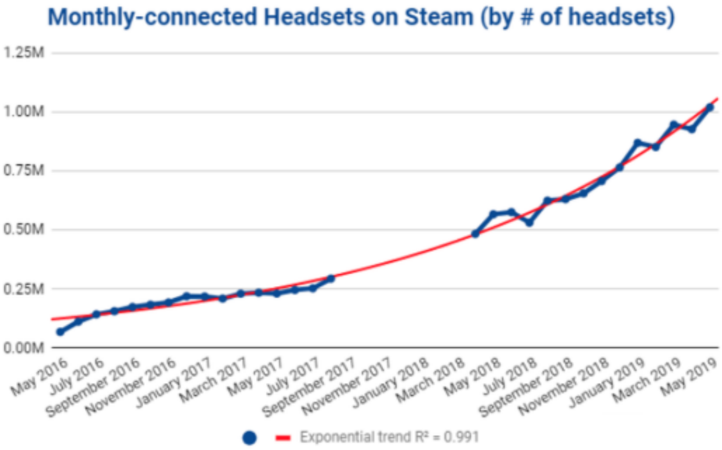

Figure 2.15: Monthly-connected Headsets on Steam (by # of headsets)

Virtual World The virtual world can be described as a collection of virtual objects in a virtual environment, the rules and the relationships governing those objects.

Immersion Immersion is possibly one of the more characteristic elements of VR and the one that mostly distinguishes it from other technologies. It generally refers to the emotional or mental state of feeling involved in the experience. This involvement can be both mental (being deeply engaged; suspension of disbelief) and/or physical (bodily entering into a medium, synthetic stimulus of the body's senses via the use of technology).

It closely relates with the term "presence", which is the illusion of being in the place rendered by VR. Slater in [86] splits the concept of presence into place illusion (PI) - The user feels he is in the scene although cognitively he knows he's not there - and plausibility illusion (Psi) - The user feels the scene is credible, although cognitively he knows it's not true.

In order to achieve PI the VR should be perceived through natural sensorimotor contingencies, based on an active vision paradigm [31], this means, we should perceive the virtual world by using our own body, by performing actions like head turning, leaning, reaching, looking around and so on.

On the other hand, Psi requires that the environment responds accordingly to actions of the participants and that it generates spontaneous actions towards them (e.g when the environment includes virtual human characters, these should respond to the presence and actions of the users) .

When both PI and Psi operate, users will be likely to feel immersed in VR.

Sensory Feedback Sensory feedback is an essential ingredient to virtual reality. The VR system should provide direct feedback to the users based on their actions. The basic example of this behaviour happens with the visual sense that should receive feedback every time the user position changes. More advanced VR experiences can also provide haptic (touch), sound or smell experiences.

The body is a focal point at which PI and Psi are merged [86]. The action involved in looking at your own body and seeing your actions reflected in it provides a very powerful sense of ownership. This is the result of the correlation between proprioception (the ability to sense the relative positions of parts of our bodies and the amount of muscular effort being involved in moving them) and visual exteroception (collection of information regarding environmental characteristics) and results in the increase of both PI (your body is where you think it is) and PSI.

Interactivity Jonathan Steuer in [90] defines Interactivity as the extent to which users can participate in modifying the form and content of a real-time mediated environment; it depends on speed - the rate at which user actions are reflected in the virtual world - range - number of possible results for a user action - and mapping - a system's ability to map its controls to changes in the mediated environment in a natural and predictable manner.

These concepts are common along the different types of VR systems [94]:

- Cave Automatic Virtual Environment (CAVE) - Uses rear-projections screens, each driven by one of a set of coordinated image-generation systems. It brings wider field of view and the ability to give a shared experience to a small group of users at cost of higher financial costs, higher space requirements and brightness, contrast and color limitations.
- Head-mounted VR (HMD) - Devices worn on the head or as part of a helmet that typically contains two optic displays (one for each eye) which stream data as a stereo scene to the user, from the perspective of each eye.
- Desktop VR

.

2.4.3 Visualization in VR

The visualization of scientific data and phenomenons typically involves high-dimensional structures, represented in three-dimensional (3D) structures. The shapes and relations between the 3D structures are often extremely important. The work of Hubona et al. [45] suggests that user's understanding of a 3D structure improves when they can manipulate the structures. VR can be used to display and manipulate those structures, providing spatial and depth cues that no other platform can. This allows rapid and intuitive exploration of the data [52]. When compared with traditional 3D visualization methods, VR provides easier navigation, more natural interactions, improved spacial awareness, smoother collaborations and more raw pixels which allow the users to make better use of the peripheral vision [101].

The proper employment of VR techniques within computational steering environments, tools that provide the ability to easily adjust simulations, can revolutionize the way the data are visualized. Figure 2.16 shows the visualization pipeline with VR integration.

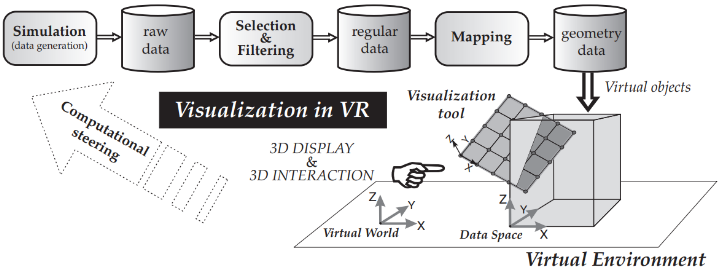

Figure 2.16: Visualization pipeline with a VR component

2.4.4 Hardware

VR hardware should be able to provide the sensation of space and depth to the users. The human visual system interprets the depth in images using both physiological cues (accommodation, convergence, binocular parallax, and monocular motion parallax) and psychophysical cues (retinal image size, linear perspective, texture gradient, overlapping, aerial perspective shading and shadows) [69]. In regular 3D graphics, monocular depth cues such as perspective, shading, shadows and texture gradients are the ones often used. Whereas, in immersive virtual reality, stereo display and head tracking are used to provide the binocular and motion parallax.

Stereo display (Fig.2.17) corresponds to the strategy of displaying a separate image for each eye. This process can be performed with many techniques, being the more popular ones, active stereo and passive stereo. In active stereo, both images are projected alternatively at a high frequency and at the same time the glasses obscure the light directed to one of the eyes. As a result, each eye of the user only sees the image intended to that eye. While in passive stereo, the two images for the left and right eyes are projected on a metallic screen by two distinct projectors with polarized filters mounted on them. By using polarized glasses of the user ends up seeing the image from one projector in one eye and from the other projector in the other eye [52].

The head tracking is used to simulate motion parallax by controlling the viewpoint on the virtual world accordingly to the spatial position and orientation of the user's head.

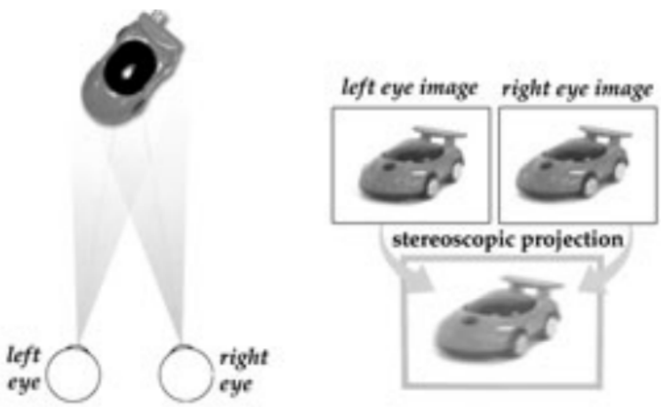

Figure 2.17: Monthly-connected Headsets on Steam (by # of headsets)

Headset	Platform	Positional Tracking	Controllers	Controllers Positional Tracking	Optics	Refresh Rate	Resolution	Price (in Jul/2020)
HTC Vive Index	PC	✓	✓	✓	130º	90 Hz	2880x1600	1100€
HTC Vive Focus	Standalone	✗	✓	✗	110º	75 Hz	2880x1600	800€
HTC Vive Cosmos	PC	✓	✓	✓	110º	90 Hz	2880x1700	700€
Oculus Quest	Standalone	✓	✓	✓	100º	72 Hz	2880x1600	500€
Oculus Go	Standalone	✗	✓	✗	101º	72 Hz	2560x1440	140€
Oculus Rift	PC	✓	✓	✓	110º	80 Hz	2560x1440	490€
Samsung Gear VR	Android	✗	✓	✗	101º	60 Hz	2560x1440	100€
Google Daydream	Android	✗	✓	✗	90º	60 Hz	2560x1440	60€
Sony PlayStation VR	PlayStation	✓	✓	✓	100º	90-120 Hz	1920x1080	200€

Table 2.1: Headsets Comparation

2.4.4.1 Head Mounted Displays

The classical way to provide the aforementioned cues is through Head Mounted Displays (HMD). An HMD displays images, one for each eye, forming a stereo scene. Each image is calculated and provided separately with the correct perspective of position of each eye with respect to a mathematical description of a three-dimensional virtual scene. Generally they can be split into two categories: high-end HMDs which offer a comfortable user experience with an independent screen screen, complex device structure and advanced technology and mobile-based HMDs that have a simpler structure and are dependent entirely on a smartphone to display stereoscopic animations in VR [105].

There are many VR headsets available on the market, according to [22] and [1], some distinguishing features are:

- Tracking - Having six degrees of freedom (6DoF) (positional tracking - user can move in space) or only three degrees of freedom (rotational tracking - user can only rotate).
- Refresh Rate - The number of times the display updates with new images (higher refresh rates result in smoother experiences).
- Latency - The time interval between the simulation and response (should be less than 20ms).
- Persistence - The time interval to switch between states, pixel on or of (should be less than 3ms).
- Resolution - The number of distinct pixels in each direction (at least 1000x1000 per eye).
- Optics - The wideness of the field of view, the ability to calibrate (focus) and the comfort of the eyebox.
- Power Supply - The device works as a standalone device or is powered by a PC or mobile.

A summary of some of the most popular headsets of today and a subset of the aforementioned features in addition to price and the existence of controllers with 6DoF is available on figure 2.1.

At the time of writing, the VR industry seems to be trending towards having positionally-tracked headsets with positionally-tracked controllers.

Feature Name	Chrome	Firefox Reality	Oculus Browser	Samsung Internet
WebXR Core Module	Supported Chrome 79+	Supported	Supported 7.0+	Supported 11.2+
WebXR AR Module	Experimental	Supported	Not Supported	Supported 11.2+
WebXR Gamepads Module	Supported Chrome 79+	Supported	Supported 7.1+	Supported 11.2+

Table 2.2: Support Table for the WebXR API

2.4.4.2 Projection-based Displays

This technology displays the stereo images via projectors, compared to HMDs it provides a larger field of view, with higher resolutions, and usually better image quality. It also allows multiple users to be immersed at the same time. It is the technology used by CAVE-like systems. This way to visualize VR provides a complete surrounding projection.

2.4.5 Software

Previous sections defined what VR is and what hardware is typically used. Next sections will try to describe how to develop for VR these days.

2.4.5.1 WebXR

We will start this section by describing what are the options when deciding to develop VR applications for the web.

WebXR is the API that allows the development of VR and AR Web applications. It supersedes the previous standard: WebVR. Both WebXR and WebVR were invented with the intention of simplifying the development and experience of VR applications in web browsers [92]. In comparation to WebVR, WebXR combines VRand Augmented Reality into a single API, supports 6 degrees of freedom tracking, has a better management of VR controllers and shows rendering improvements [9]. WebXR applications are built to be seen and reproduced by any VR headset and by working on browsers, where the WebXR API is supported. It allows the users to access applications without the need to install any additional software. At the time of writing, the support for the the WebXR device API is represented in table 2.2. The WebXR Gamepads Module is for accessing the state of buttons, triggers, thumbsticks, and touchpads associated with VR(VR) and augmented reality (AR) motion controllers on the Web. The WebXR Augmented Reality Module extends the WebXR Device API to expose the ability to create a basic augmented reality (AR) session. The WebXR Core Module covers what WebVR used to provide (managing the selection of output devices, render the 3D scene to the chosen device at the appropriate frame rate, and manage motion vectors created using input controllers).

Frameworks To facilitate and quicken the development of applications, typically developers resort to frameworks. There are only few WebXR ready frameworks existent today: A-Frame and React360. This frameworks eliminate the complexity of dealing with WebXR directly and allow a more efficient use of JavaScript code. A brief comparation between this frameworks, based on [49] [53], will happen in this section.

A-Frame[2] is an open source framework able to create VR experiences with HTML. It's based on Three.js[3] and encapsulates the complicated WebGL and JavaScript code into HTML. It is seen as the most beginner friendly framework available for web developers of VR. A feature that sets A-Frame apart from many other frameworks is its visual inspector (Fig.2.18). It is designed to inspect and tweak A-Frame scenes at runtime [26].

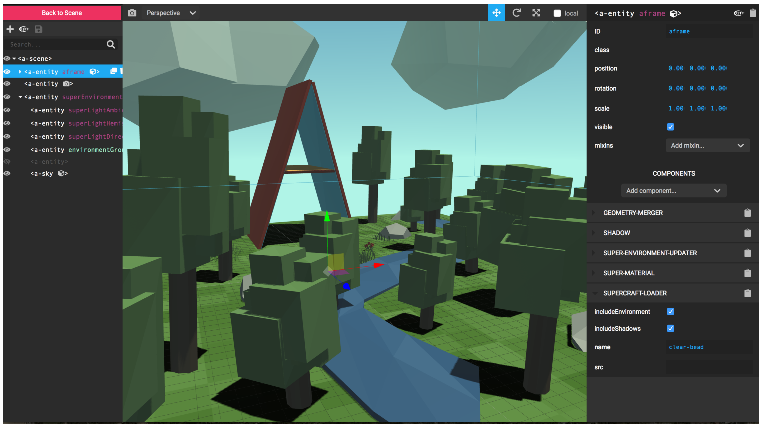

Figure 2.18: A-Frame online inspector

React 360[4] is a framework designed to create 3D and VR experiences. It is built on the top of React, a popular JS library, and that makes it preferable for websites that have many 2D and 3D elements. However if the application deals with many 3D objects, react 360 is not a fit.

More generic JavaScript libraries such as Three.js and Babylon.js[5] were also briefly considered as auxiliary tools, due to its usefulness in building animated 3D computer graphics. Three.js is a JavaScript library designed for 3D rendering. It provides developers with complete control over 3D objects and empowers them to create more complex WebXR experiences. Has an extensive documentation and the largest community in this list of frameworks. Babylon.js is a JavaScript-based WebGL library similar to Three.js. Originally created for game development, is a full-featured framework, with a dedicated testing playground and beginner-friendly abstractions. It it is fully capable of creating WebXR experiences.

[2]https://aframe.io/
[3]https://threejs.org
[4]https://facebook.github.io/react-360/
[5]https://www.babylonjs.com/

Game Engine	Compatibility Steam VR	Compatibility Oculus VR	WebXR Compatibility
Unity	Yes	Yes	Limited
Unreal Engine	Yes	Yes	No

Table 2.3: Game Engines Compatibility Comparation

2.4.5.2 Game Engines

If the web is not the main target, one should consider the use of game engines when looking for ways to develop VR applications. For this purpose, both Unity3D[6] and Unreal Engine[7] will be briefly described with focus on VR as the end target. Unity3D is a development engine that was first released in 2005 and supports three development languages including C#, UnityScript and Boo. Unreal Engine has a long history as a game engine dating back to 1998, is an open-source engine and only supports C++. Both Unity and Unreal Engine have assets markets that allow purchase of pre-made 3D models, objects and environments. Both have an extensively developed documentation on how to use the engines and on XR development. In terms of graphics department, Unreal Engine 4 has a few advantages over Unity. When it comes to projects that use these tools, C4X Discovery uses Unreal Engine 4 to visualize molecular data in VR [35] and Audi, a german automobile giant, developed a modular VR training with Unity, for example.

One concern with the use of game engine refers to the compatibility with different headsets. The development of VR apps for specific headsets relies on the use of third-party software development kits (sdk) within the game-engine. By default Oculus SDK works only for Oculus devices; targeting Open VR SDK (Steam VR) instead should work for headsets of different brands (Oculus, Mixed Reality, Vive, etc). Table 2.3 summarizes the compatibility of the studied game engines.

2.4.6 Interaction Techniques

In this section, we will describe the existing techniques to provide basic interaction for virtual environments, as well as some typical problems with them. We can divide interaction techniques into two categories: navigation and interaction with objects.

2.4.6.1 Navigation

For most scientific visualizations, there is no natural navigation scheme, especially for worlds where you don't normally walk or fly through. This makes choosing the right navigation technique a crucial decision. There are many different techniques of navigation to be considered:

I Real Walking

In 1999, Fred Brooks et al. [100] pointed out that physical movement powerfully helps the illusion of presence. Real walking allows you to feel kinesthetically how big the spaces are making it better

[6] https://unity.com/
[7] https://www.unrealengine.com/en-US/

than virtual navigation or other hybrid methods, in terms of place illusion and plausibility illusion. In the paper, it is reported that real walking provides a strong sense of presence, it is easier to use and improved the efficiency and spatial awareness of the space explored. In addition, in this case, as there is no conflict over signals received by the visual and vestibular system, it is less likely to cause *cybersickness*[2.4.8].

The biggest problem with the real working method is the limitation of the physical space in which the user is. First, it is difficult to make assumptions about how much space the user would have available when using the application. And secondly, as users are encouraged to use their bodies as naturally as possible to explore the virtual world, there might be some health and safety concerns since from the user's perspective, the real world is completely invisible.

II Redirected Walking

As stated in [78], redirected walking captures the benefits of real walking while extending the possible size of the virtual environment. This method works by interactively rotating the virtual scene about the user. The user does not notice this rotation because the algorithm exploits the limitations of our human perceptual mechanisms ability to detect position, orientation and movement.

Basically, this method manipulates users to believe that they are walking along a straight line, where in physical reality they are walking along a curve. In extreme situations, users can be manipulated to walk along a large circle in the physical world repeatedly, thinking that they are walking along an infinitely long straight line in the virtual world.

This method requires a lot of programming and very careful design of tasks in the virtual environment.

III Walk in Place

Walk in place is another way of navigating VR and situates between physical and virtual navigation. This method encourages the user to involve the whole body in the realistic walking movement as much as possible, but without actually moving forward. Users should be able to explore the virtual environment with their body, such as bending it to look under the table. But they must also be able to walk in the virtual environment imitating the movement of walking without physically advancing. The technical implementation may vary, but in most cases, head rotation data is used to define the direction of travel, which is quite straightforward. The least direct part is determining the speed of the trip, or simply whether the user is actually walking or not [100].

IV Virtual Navigation

The most extreme version of virtual navigation is to use a joystick or touchpad to control the direction and speed of travel.

However, in VR, this usually causes a strong feeling of nausea. There are two reasons: first, in real life, most of the time we would be looking in the direction of travel. Therefore, if we use a joystick to define the direction of travel and that is not the actual direction of my head, it would generate a conflict in our sensory system that would cause *cybersickness*. Second, this method generally produces many changes in speed or acceleration, which is another factor that has proven to be a major contributor to *cybersickness* as well. To avoid this, the simplest solution is to use the user's head direction as the direction of travel at a certain speed.

V Teleportation

Another way to travel in VR is a method called teleportation. With teleportation, users can travel from one place to another, looking at the new location, selecting it and the next moment they are in the new position. The new location that users want to travel to is often called the target location. In some applications, multiple target locations are predefined; users can then look at or point to a target with the controller to indicate where they want to travel to. In other applications, instead of predefined targets, users can teleport anywhere on the ground, looking or pointing to an arbitrary position. In both cases, it is important to provide some type of feedback to the user to confirm their selection of the destination location.

To reduce the feeling of disorientation that can occur when using this method we can instead of just repositioning the virtual camera from one frame to the next, generate some of the in between frames (preferably blurry) or we can give to the user another visual cue (an avatar for example) to get him/her used to the new position better. This last strategy uses the fact that we are subconsciously constantly trying to find out what other people around us can see [8].

VI Move The World

A different perspective to the navigation problem is the world travel paradigm. The user is allowed to grab objects in the world (the data) via a controller and move it as desired. It offers several advantages compared to the previous strategies, given the perspective of a scientific tool as the objective: Since the user is always consciously manipulating, it is almost impossible for users to lose the data object, they can place it wherever they choose to and the feedback of watching the data as it is being moved looks natural and intuitive.

2.4.6.2 Interactions

Our ability to interact with objects in the virtual world is essential in creating the plausibility illusion in VR. Just like with navigation there are different ways we should control objects in the virtual world. Follows a couple of strategies that are commonly used, based on the content acquired in an online course provided by University of London & Goldsmiths, University of London [71]:

I Objects Within Reach

The most naturalistic way to interact with an object within reach is to use a simple virtual hand to represent the user's real hand. Assuming that our controllers are tracked with six degrees of freedom in VR, the movement of the virtual hand can be a direct mapping of the user's real hand. Thus, we can move our hands and reach objects exactly the way we interact with real objects around us.

In this method, a strategy to selection can be when the hand overlaps an object in the 3D world, we can press a button to select the object and then this object will be attached to our hands, so that we can rotate it or take it to a different position.

The entire selection and manipulation process is very intuitive, as it directly simulates how we interact with real objects in the real world.

There are few problems with the simple approach of the virtual hand, the biggest problem is that users can interact only with objects within their reach. If they want to reach something further, they must first approach the object, which comes with its own problems and restrictions. Second, as the interaction is very naturalistic and intuitive, users also have high expectations and can be frustrated when things don't meet their subexpectations.

II Hyper-Natural Interaction

Hyper-natural interaction refers to the act of extending users' natural movements, giving them "superpowers" (for example, we can extend the user's arms).

A more careful and very famous attempt to generate the illusion of arm extension is the go-go technique created by Poupyrev in 1996 [77]. It defines a radius around the user around two thirds of the physical length of the arm, an area where we are very sensitive. and familiar with. And within that radius, the user has a direct individual mapping of his physical hand to the virtual one. So, I'm going to interact with objects within that range, just like I do in real life. But beyond that radius, the virtual arm is designed to be mapped non linearly to the real arm, which allows the virtual arm to be further away than the real arm.

Overall, this technique is very intuitive and easy to learn. And, unlike other methods that allow the user to raise their arms infinitely, this method imposes restrictions on how much and how quickly we should raise our arms. These restrictions reflect our natural ability to deal with objects at a distance and thus prevent this superpower from getting out of control.

III Magic Interaction

Magic interaction it's often referred to as ray casting or virtual pointer. This method, allows the user to hold the controller as a laser pointer. He/She can then can use it to indicate the object of interest and confirm the selection (by clicking a button).

Although good for selection this technique can be quite difficult to master in terms of manipulating objects out of reach. A couple of strategies try to solve those problems: When an object is out of reach we can use **close manipulation** to bring the object near to the user; **Popping** allows the user to bring the object into his hands and after manipulation the object goes back to its original position; **Copying** creates a copy of the object near the user and mirrors the manipulations of the copy to the original. If this strategies of bringing the object to within reach are not an option, there are still strategies to remotely manipulate them: the **slave method**, makes the object follow the translations of the pointer; The **stick method** works similarly but besides translations it also allows rotations [52].

2.4.7 Full-Body Tracking

Being able to track more parts of the user body provides a more natural experience. Originally, systems for sensing people's entire bodies required special purpose hardware, typically expensive, such as full-body suits bundled with multiple sensors. However, recent algorithmic advances in computer vision and the simultaneous increase of computational power available on personal computers allow this detection systems to be vision-based [98].

Vision-based position and pose tracking can be achieved by collecting 3D data from a stereo depth-based camera and correlate it with pre-defined, susceptible to calibration, 3D stereo models of a human forms, typically via neural networks techniques. The result of this method typically needs to be mapped from raw 3D information to virtual controllers so that they can be used by applications.

2.4.8 *Cybersickness*

According to [54], feeling discomfort as a side effect of using VR systems has been one of the biggest threats to the widespread adoption of technology and therefore VR developers must take special care to its users security and well-being.

The term used to cover any disease associated with VR is *cybersickness*. VR users can fell cybersickness when the information gathered (typically visual) is not is not what the body expects to receive.

Common symptoms of cybersickness are:

- nausea - users can start experiencing unpleasant sensations in any of the following: throat, esophagus, stomach or upper abdomen;
- dizziness - users can feel a sense of movement, such as turning, even after the stimulus is removed;
- drowsiness - users may become less alert, yawn, and eventually start to fall asleep;
- cold sweating - users begin to sweat or increase their sweat, but not in response to increased ambient temperature;
- pallor - users experience a whitening or loss or normal skin color in the face;

- headache - users develop headaches that may gradually increase in intensity and remain long after use;
- fatigue - users may become tired or exhausted after a long experience;
- eyestrain - users may feel that their eyes are tired, fatigued, sore, or aching.

It is hard to correlate specific actions (causes) to specific symptoms from the above list (effects) due to its wide variability between among different users. Even so, a set of recommended strategies aiming to reduce the likelihood of those effects happening is listed below:

- Avoid eye convergion (objects moving towards the camera).
- Use darker backgrounds.
- Avoid focusing on different depths.
- Avoid sudden brightness changes
- Avoid sudden acceleration or deceleration.
- Keep the framerate larger than 30fps (frames per second).

2.4.9 Future directions

The doors to benefit from VR immersion are open but there are still some challenges ahead [29] [11].

In terms of technology the general goal is to make **latency** get down to acceptable levels and keep it there within a **reliable** range so that users can enjoy a good and reliable VR experience and minify its common side-effects. This challenge is closely related with the various types of delays involved in VR systems. There are both computing and communication delays. Heavy image processing (usually >1M polygons in real time) requires high computational power which might not be available in the local HMD graphics processing unit (GPU). Offloading this task to remote servers relieves the computation burden at the cost of higher communication delays (typically is still worth it in VR systems or MR and AR systems with low resolution). The communication delays are a consequence of the network speed and distance between end users and the computing servers. It is important that after achieving the desired latency, the system should be able to keep it without lag spikes and dropouts. Failing to do so will result in a more detached user experience.

2.5 Summary

In this chapter we described the state of the art of each one of the areas related with this dissertation.

We started by setting the context of this dissertation by explaining computational biophysics in general and computational neuroscience in particular. We explained computational modelling and computational simulations with the help of concepts like level of analysis and level of detail. Later we listed a set of computational neuroscience tools either for visualization or simulation and a few conciliating both. We finalized the biophysical section by providing some future directions for the evolution of the field.

Then, we moved to the visualization section where we introduced the concept of visualization and 3D visualization. We provided basic concepts of 3D graphics, described current technologies of the field and identified common problems with interactive 3D visualization tools.

Afterwards we moved to the main topic addressed in this chapter: virtual reality. Here we explained the technology at its core, the interaction techniques and the advantages of using VR for visualization, among other contextualization topics like the necessary hardware for VR, a comparation between popular head mounted displays, current protocols and frameworks available, as well as recommendations to avoid *cybersickness* and future directions of the field.

In the next chapter we will compare the technologies and techniques described here, in an impartial and theoretical way and explain in detail the ones we consider the best for the needs of this dissertation.

Chapter 3

Methodology

This section is intended to address, from a more theoretical point of view, the various aspects related to VR and computational biophysics techniques that were used in the present study.

In building a visualization prototype of biophysical data with immersive capabilities based on VR, there are two tasks that stand out:

- the user's movement mode,
- how to interact with the objects on the scene.

The interaction component combines several methods and technologies in order to provide a sensation as close to the real as possible. This sensation is the main factor by which the user might benefit from performing actions in the VR environment. However, for the prototype to have value for the partner company, it is necessary to resolve several issues of compatibility with the company's business strategies. Each one of them is presented, together with the respective implemented solution in section 3.4.

On the other hand, computation biophysics in general and scientific visualization in particular face their own specific problems: on a time when the amount of data to be collected reaches levels never seen before, we must take that into account when creating such visualization platform. The approach chosen and the reasons behind that choice are presented in section 3.3.

3.1 Context & Objectives

Before diving in each one of the main areas of this thesis in particular, we will first address, in this section, how we expect to integrate them and what are the requirements for the prototype to be developed in parallel to this document.

Nowadays, typical tools for visualizing biophysical data, also allow the users to quickly run and steer simulations. The process of using this tools, by researchers, commonly contains the following steps:

1. Define the parameters that a given model allows them to tweak (Fig.3.1).
2. Run a simulation with those parameters (Fig.3.2).
3. Visualize the results of the simulation in an interactable standard 3D scene along with 2D plots (Fig.3.3).
4. Repeat.

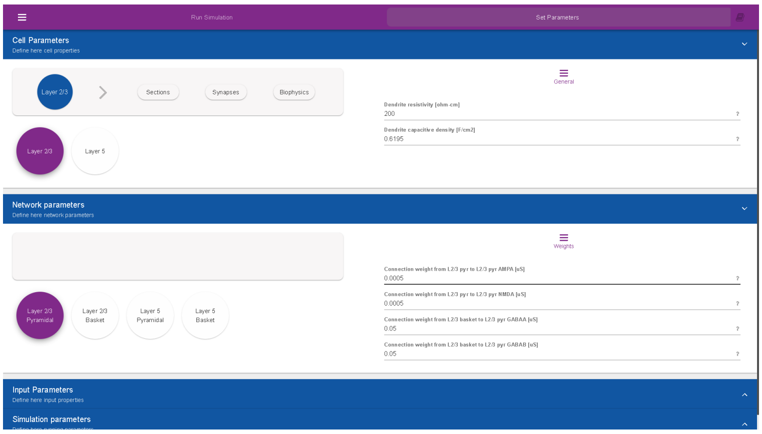

Figure 3.1: Example of biophysics visualization application (HNN[68]): Setting model parameters.

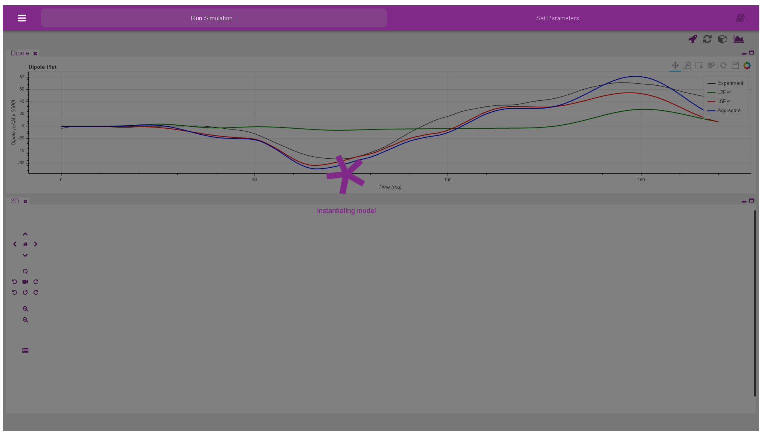

Figure 3.2: Example of biophysics visualization application (HNN[68]): Running simulation.

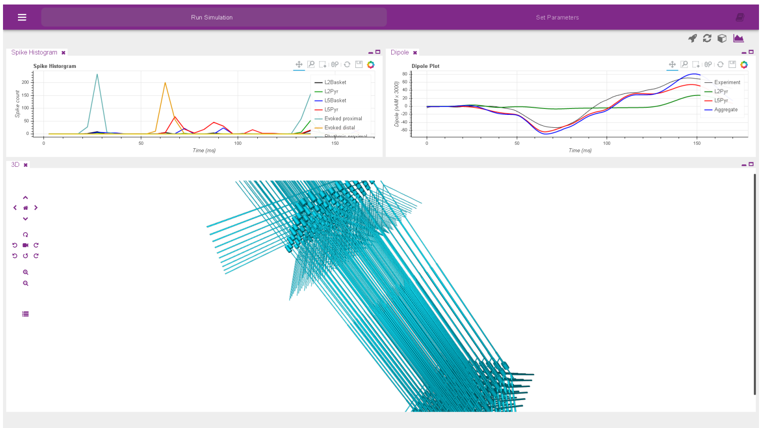

Figure 3.3: HNN[68]: Simulation results.

With this dissertation, we want to evaluate if replacing the standard 3D scene with a scene with VR capabilities would be a valuable change for the researchers of the field of computational biophysics. To do so, we defined a set of features that our prototype should implement (Fig.3.4): The prototype should be able to read a declarative biophysical model and generate VR scene showing the 3D structures defined in the model. Allow the user to interact with those structures, and as proof of concept, allow the running of a basic, potentially mocked, simulation. An extra requirement is that the result should be available in the form of web application.

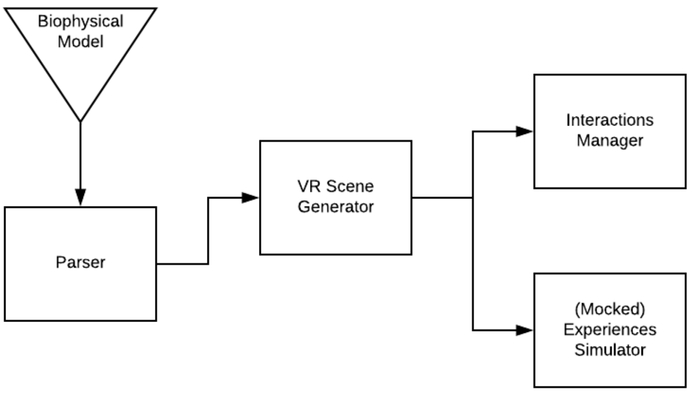

Figure 3.4: Block diagram with the layers / features to be implemented in the prototype.

It was agreed with the company that primary focus of the prototype should be the interaction with the visualized scene, so we defined a set of concrete goals regarding that area:

- Given a geppetto model generate a VR scene with the 3D model and be able to interact with it (and its sub-parts individually) by rotating, scaling and selecting.
- Generate an interactive GUI, where users can have custom actions (e.g. run simulations, apply colormaps).

Besides meeting the aforementioned criteria, a good result should also provide the user with an immersive experience.

Being successful in achieving these goals will result in an application with great value for the field. Users of this application would be able to load a biophysical model, fill its parameters, run simulations and visualize the results.

The biggest distinguishing factor of the aforementioned application when compared to current computational biophysics tools is the way users can interact and evaluate the results of the simulations, since you would be immersed in a virtual reality scene and not simply using a standard screen.

This visualization/interaction contemplates the navigation of the user in the VR scene with the simulation results using real body rotations and a joystick, the ability to easily and intuitively select any object in the scene with the help of a raycaster projection originating in the user's hands and the ability to manipulate the objects in a natural way, this manipulation involves both rotations, scaling and translations with the first two being done through hand gestures.

We expect that this difference will have a very big impact on scientific research, namely facilitating the interpretation of data and stimulating new discoveries.

3.2 Project Specification

This section contains the specification of the actors, their user stories and supplementary requirements, serving as agile documentation of project requirements.

3.2.1 Actors

Our Actor ("User") models a generic user/researcher that has access to all the functionalities of the application developed, such as defining model parameters, running simulations and visualize and interact with data.

3.2.2 User Stories

For our application, we consider the user stories that are presented in table 3.1. These user stories aim to define the project requirements and anticipate the subjects to be discussed in this chapter.

The first 5 user stories clearly identify the 3 key concepts of computational biophysics (modelling, simulation and visualization), while the last 5 describe the basic actions typically required in an immersive VR system (navigation, selection and manipulation).

In detail, user stories 'US01' and 'US02' encapsulate the process of computational modelling - creation of mathematical models in computational science that requires extensive computational resources to study the behavior of a complex system using computer simulations - addressed in 3.3.1.

User story ’US03’ confines the process of computer simulations itself - process designed to predict the behaviour of or the outcome of a real-world or physical system modelled with computational models - addressed in 3.3.2.

User stories ’US04’ and ’US05’ crave for what we call the visualization process - technique of creating images, diagrams, or animations to facilitate the interpretation of both abstract and concrete data - referred in 3.3.3, with the nuance of being performed in virtual reality.

The remaining user stories refer to specific sub-functionalities of the visualization process using Virtual Reality, namely navigation in VR (3.4.1) in ’US06’, selection in VR (3.4.2.1) in ’US07’ and object transformations in VR (3.4.2.2) such as: rotations, scaling and translations, respectively user stories ’US08’, ’US09’ and ’US10’.

Identifier	**Priority**	**Description**
US01	medium	As an User, I want to define/load a computational model, so that I can test different complex systems
US02	medium	As an User, I want to define the variables of my computational model, so that I can test different complex systems with different configurations
US03	medium	As an User, I want to run simulations, so that I can check the impact of my parameters definition and formulate new hypothesis
US04	high	As an User, I want to visualize the morphology in study using VR, so that I can have a natural, immersive and intuitive experience when analyzing it
US05	high	As an User, I want to visualize the simulation results using VR, so that I can have a natural, immersive and intuitive experience when interpreting simulation data
US06	high	As an User, I want to move freely around the visualization scene using my body, so that I can have a natural way to explore the scene
US07	high	As an User, I want to be able to select scene objects both near and far from me, so that I can collect individual detailed information or perform individual actions
US08	high	As an User, I want to be able to naturally rotate scene objects, so that I can explore the whole object with ease
US09	high	As an User, I want to be able to naturally scale scene objects, so that I can analyze the object with the size that better fits my needs
US10	high	As an User, I want to be able to easily change scene objects position, so that I can analyze the object without having to move towards it

Table 3.1: User Stories

3.2.3 Technical requirements

Technical requirements are concerned with the technical aspects that the system must meet, such as performance-related issues, reliability issues and availability issues. Those are described in table 3.2.

Identifier	Name	Description
TR01	Accessibility	The system must ensure that everyone can access the pages, regardless of whether they have any handicap or not, or the Web browser they use
TR02	Usability	The system should be simple and easy to use
TR03	Performance	The system should have a frame rate higher than 30 frames per second to ensure the user's welfare
TR04	Web Application	The system should be implemented as a Web application with dynamic pages (HTML5 and JavaScript)
TR05	Security	The system shall protect information from unauthorized access through the use of HTTPS
TR06	Robustness	The system must be prepared to handle and continue operating when runtime errors occur
TR07	Scalability	The system must be prepared to deal with the growth in the number of users and their actions
TR08	Ethics	The system must respect the ethical principles in software development
TR09	Interface Provider	The React framework must be used
TR10	Model Interpreter	The Geppetto framework must be used

Table 3.2: Technical requirements

3.3 Computational Biophysics

The conditions related to computational biophysics referred in the state of the art chapter, led to the search for efficient solutions, not only thinking about the implemented prototype (which aims to be a showcase rather than a complete tool), but also for the most common scenarios of using computational biophysics with focus on computational neuroscience.

When creating a tool for biophysical computing, it is not possible to predict the level of analysis, nor the level of detail that will be used, so possible optimizations specific to a certain level were not considered; but else the recommended methodologies for the most common scenarios in those tools: the creation of a model, the process visualizing that model and the mechanism to simulate over that model will be explained below.

3.3.1 Models

Computational models are increasingly important for studying complex neurophysiological systems. As scientific tools, it is essential that such models can be open, accessible and reproducible to the larger range of scientists. However, the publishing of models can be very fragmented, with diverse modeling approaches (LEMS[17], NeuroML[36], NWB[95]) making them inaccessible and difficult to reproduce [17]. To address this issue, we use a model abstraction provided by Geppetto[18] which decouples domain-specific modelling formats from the visualization components. This meta-model follows an Object-Oriented Programming paradigm by allowing the definition of variables (represents an instance of a given type with an initial value), types (represents the structure of an entity) and values (something that can be assigned to a variable or to a type (the default value)), supporting type inheritance (types can be extend by other types) and composition (variables can contain different types), in a declarative way.

3.3.2 Simulation

The ability to run simulations on neuroscientific models is critical for the understanding of brain functions. The strategies to run simulations obviously varies with the goal of the simulation. Our model contains agnostic data that can later be converted to simulator specific formats such as NEURON[40]. This allows the computational biophysics tools to serve specific data (concentration of ion channels, conductance, etc), to specific simulators, and those are the ones responsible to numerically simulating intracellular electrical and chemical dynamics utilizing the partial differential equations (as shown in 2.1.1.1). In our case, we allow one type of simulation which is the injection of current in the model (e.g. in the soma of the CA1 cell), the impact in all the model is measured by relying in Ohm's law (Eq.3.1) and visualized by pseudo-colouring the morphologies to show changes over the course of the simulation.

$$I = \frac{V}{R} \tag{3.1}$$

where I stands for current, V for voltage and R for resistance.

3.3.3 Visualization

The visualization of 3D structures starts by having the model defining the primitives and positions of each object in the scene or by defining exported formats like OBJ or COLLADA to achieve the same. A model interpreter (parser) is responsible to gather that information and use it accordingly, typically by creating the three dimensional objects using an external library.

Depending on the level of detail / complexity of the scene, the visualization process has to contain some fallback strategy for situations where the scene high demands can no longer be put up with at a reasonable framerate. Taking note of the amount of meshes being rendered and define a threshold for the maximum amount allowed is probably the simplest effective solution and the one that was followed in the prototype implementation. When the threshold is passed, the level of detail of each object levels down (e.g. cylinders start to be represented as lines). Other interesting technique that was considered defends that the level of detail should be based on the distance to the user (objects rather distant should use a lower level of detail than objects close by). The threshold was preferred over the distance for the prototype due to practical reasons, as we were aware that the scenes wouldn't be very distant. For larger, more realistic, projects the latter might be a better alternative.

Another typical requirement in the area of visualization refers to the interest in highlight different cell aspects at different times. This feature can be achieved easily if the model contains all the cells morphologies specified, making it just a matter of re-render the scene with the interesting meshes being displayed. Note that, by default, its is preferable to minimize the number of meshes in the scene (obviously without compromising the intent of the application).

3.4 Virtual Reality

As concluded in the previous chapter, an effective VR module must pay particular attention to both movement and interactions. For each of the two situations identified above, a strategy was chosen that seemed to be the most appropriate to give an answer to the problem in question. These strategies were: Virtual Navigation for the navigation issue, and Magic Interaction for the interaction issue.

Throughout this section both solutions are presented and explained in detail. The approach used for the navigation problem is presented and detailed in section 3.4.1 while the choice of the interactions techniques is reviewed in section 3.4.2.

3.4.1 Navigation

One of the most basic VR interactions is the user's ability to move around and explore the virtual environment. As stated in the analysis of the state of the art there are different ways one can design navigation in VR depending on what they want the users to do. For our specific case, a tool to be used by scientists to help research in Biophysics fields, we considered two factors when choosing which navigation strategy to use:

- Impact on user's performance.
- Ergonomics.

I Impact on user's performance

To assess the impact on user performance, we rely on the study performed in 2010 by Bernhard E. Riecke et al. [80]. In that study it was concluded that there is an overall benefit for full physical motion (real walking condition) as compared to joystick navigation (virtual navigation). However, when the joystick navigation is mixed with bodily rotations (real rotation condition) it provided considerable performance benefits over joystick only and almost equalled walking performance. Teleportation was not considered in the aforementioned study, and neither in our scenario, as free user movement increases the naturalness of the scene.

II Ergonomics

Another non-negligible factor is the process of designing or arranging workplaces for VR tools. If the tool requires a minimum amount of space higher than the typical amount of space dedicated to the workers, which is what happens in the case of real walking, companies might see that as a barrier to adopt this new technology.

Both factors considered we decided to opt for using joystick navigation plus real body rotation based on the headset orientation as the navigation strategy of the prototype.

3.4.1.1 Orientation

The orientation of the user might be determined with different techniques depending on the headset used. In this dissertation, we will be using Oculus Quest as example. In this case, the orientation of the user is given by the inertial measurement unit (IMU) present in the headset. The gyroscope (part of the IMU) measures the head's orientation change at a rate of 1000 times a second. This orientation is usually represented by Euler angles (yaw, pitch, roll) of the corresponding rotations in relation to the reference orientation (Fig.3.5). However the implementation is based on the concept of quaternions (a four-element vector that can be used to encode any rotation in a 3D coordinate system), since those are singularity-free (it doesn't lose one degree of freedom in a three-dimensional opposed to what happens with Euler angles operations).

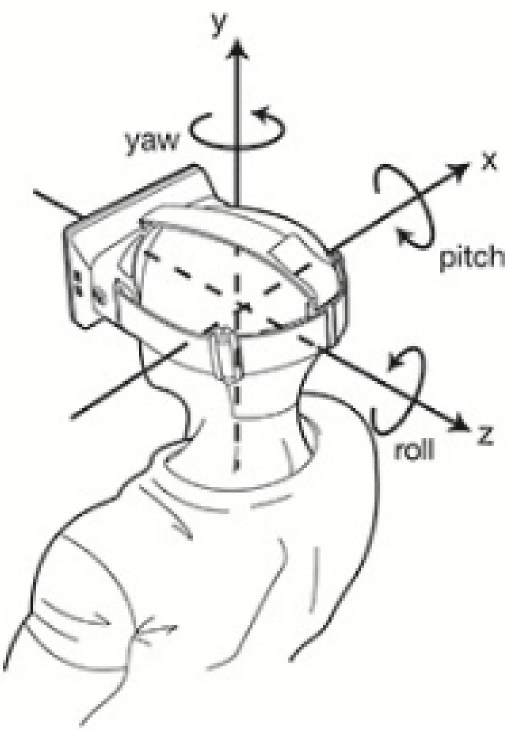

Figure 3.5: Euler angles representation

I Gyro Integration

The rotation theorem of Euler states that any 3D orientation can be produced by a single rotation about one particular axis through the origin. This axis-angle representation maps directly to the space of unit quaternions as expressed in Eq. 3.2:

$$q(v,\theta) = (cos(\frac{\theta}{2}), v_x sin(\frac{\theta}{2}), v_y sin(\frac{\theta}{2}), v_z sin(\frac{\theta}{2})) \tag{3.2}$$

in which $q(v,\theta)$ corresponds to a quaternion of one unit length that denotes a rotation of θ radians about an axis vector $v = (v_x, v_y, v_z)$.

Let $q[k]$ be a quaternion representing the orientation of the headset at moment k, $\tilde{w}[k]$ the gyroscope reading at moment k, $\hat{q}[k]$ the estimated orientation, where $\hat{q}[0]$ equals the initial identity quaternion, $l = \|\tilde{w}[k]\|$ and $v = \frac{1}{l}\tilde{w}[k]$. Since l represents the rate of rotation (radians/sec) we obtain a simple dead reckoning filter by setting $\theta = l\Delta t$, resulting in Eq.3.3:

$$\hat{q}[k+1] = \hat{q}[k] * q(v,\theta) \tag{3.3}$$

in which * represents standard quaternion multiplication. This is equivalent to simple Euler integration, but extended to the 3D rotation group.

Over time, the dead-reckoning error is expected to accumulate; we typically call to that error, drift error and it is formulated in Eq.3.4.

$$e[k] = \hat{q}[k] - q[k] \tag{3.4}$$

Drift error in the pitch and roll angles is called tilt error, which corresponds to confusion about which way is up. Drift in the yaw angle is called yaw error, which is confusion about which way you are facing relative to when you started. To handle this errors we will need other sensors besides the gyro.

II Tilt Error

In order to fix the tilt error, we will use gravity as a constant vector field of magnitude 9.81m/s^2. To do so, we will be using the accelerometer (present in the IMU). The problem with the accelerometers is that they measure the sum of gravity and the linear acceleration of the sensor Fig.3.6(a). The tilt error can be described as a rotation about an axis that lies in the horizontal (XZ) plane Fig.3.6(b).

To calculate that axis, we first need to transform the acceleration estimate from the headset axis to the earth axis by applying Eq.3.5.

$$\hat{a} = q^{-1} * \tilde{a} * q \tag{3.5}$$

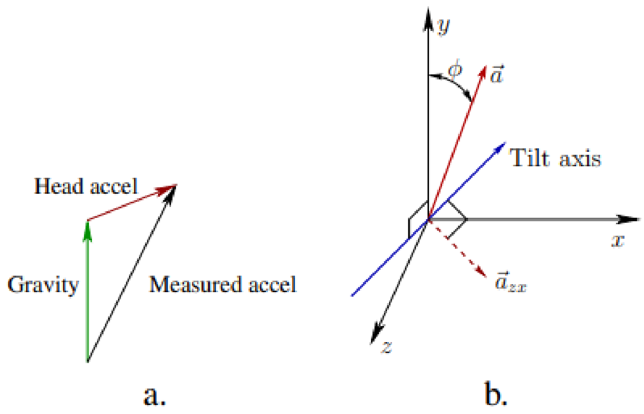

Figure 3.6: (a) Accelerometers measure sum of gravity and linear acceleration (b) To determine tilt error angle ϕ, the tilt axis is calculated, which lies in the horizontal (XZ) plane.

Then we project the result into XZ plane to obtain $(\hat{a}_x, 0, \hat{a}_z)$. The tilt axis will be normal $t = (\hat{a}_z, 0, -\hat{a}_x)$. The tilt error θ is the angle between $\hat{a}$ and the vector (0, 1, 0). This process is used as a long-term complementary filter to the short-term readings of the gyroscope, because the averaged accelerometer output, over a long time, produces a good estimate for the direction of gravity. The final transformation is formulated in Eq.3.6 where α is a small gain constant and t is the tilt axis.

$$\hat{q}'[k] = q(t, -\alpha\phi) * \hat{q}[k], \tag{3.6}$$

III Yaw Error

Regarding the yaw error, it corresponds to the rotation about the vertical axis (parallel to the gravity vector). To correct this issue we will rely on the measurement of the magnetic field using a magnetometer (included in the IMU). Such as the accelerometer, the magnetometer also measures the sum of several sources, and it is prone to be triggered in different situations (circuit boards in the headset or indoor magnetic materials) and not only by the magnetic field of earth (which by itself is very fluctuable).

To avoid this fonts of uncertainty, the user is required to calibrate the headset regularly so that it can eliminate the offset of local field interventions. Given a magnetic measurement, after calibration, $\tilde{m}$, and the correspondent orientation $\tilde{q}$ and letting $\tilde{m}_{ref}$ be a magnetic value observed earlier (before drift errors) and $\tilde{q}_{ref}$ the orientation that corresponds to $\tilde{m}_{ref}$. We start the correction by bring both readings to the earth referential with Eq.3.7.

$$\tilde{m}' = \hat{q}' * \tilde{m} * \hat{q} \quad and \quad \tilde{m}'_{ref} = \hat{q}'_{ref} * \tilde{m}_{ref} * \hat{q}_{ref} \tag{3.7}$$

If the values above calculated differ significantly, there is yaw drift and we need to apply Eq.3.8

$$\hat{q}'[k] = q((0,1,0), -\alpha_2(\theta - \theta_r)) * \hat{q}[k] \tag{3.8}$$

where $\theta = atan2(\tilde{m}_x', \tilde{m}_z')$, $\theta_r = atan2(\tilde{m}_{refx}', \tilde{m}_{refz}')$ and α_2 is a small gain constant.

Figure 3.7 compares the impact of the corrections described.

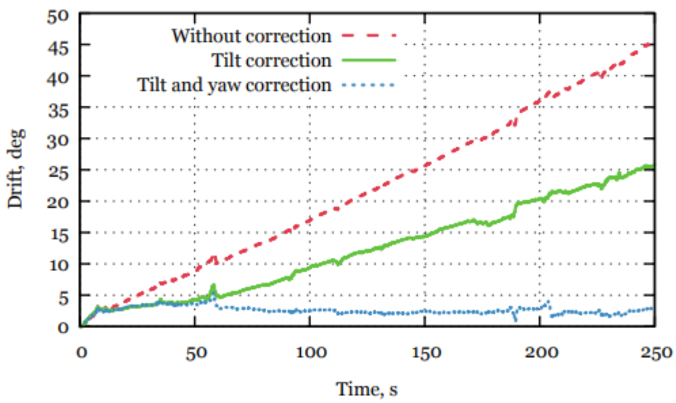

Figure 3.7: Effect of the correction strategies over time in terms of drift degrees

3.4.1.2 Joystick Movement

When it comes to the user's movement, we opted for directly adapt what people have been doing with the 2D game controllers into VR where the users use a joystick to control both the direction and the speed of travel. We did this because it seemed to be a good compromise between user's performance and ergonomics. To make the process seem more natural, the joystick inputs are applied relatively to the orientation of the user's head and not absolute (this means that, if the user looks up and moves the joystick forward, the user will move forward in the virtual world). The whole movement process consists in three steps: update the velocity, get the correct movement vector, update the position.

I Update Velocity

Let the joystick (thumbstick) input, j, be translated by 2 values, x and y, which represent the horizontal and vertical movements respectively in the referential centred in the joystick center.

Each of those values can be tilted until the max value of 1. Let the current velocity v_c be represented as well with 2 values, x and y, also to describe the horizontal and vertical components respectively. The new velocity v_n can be calculated with Eq.3.9.

$$v_n.x = v_c.x + (\alpha * \Delta t * |j.x|) \qquad and \qquad v_n.y = v_c.y + (\alpha * \Delta t * |j.y|) \tag{3.9}$$

where α represents the default acceleration (m/s) and Δt the time in seconds since the last update.

II Get Movement Vector

Let $\vec{o}$ be the orientation vector of the user's headset, and $\vec{m}$ the movement vector we want to calculate. We can create an Euler vector, $\vec{r} = (\vec{o}.x, \vec{o}.y, 0)$ and apply a rotational transformation by rotating $\vec{m}$ on its various axes in the specified, r amount per axis. This transformation is internally performed using quaternions in order to avoid singularities. To convert from Euler (ϕ, θ, ψ) to quaternion we can use Eq.3.10.

$$q(a,b,c,d) = \begin{pmatrix} cos(\frac{\phi}{2})cos(\frac{\theta}{2})cos(\frac{\psi}{2}) + sin(\frac{\phi}{2})sin(\frac{\theta}{2})sin(\frac{\psi}{2}), \\ sin(\frac{\phi}{2})cos(\frac{\theta}{2})cos(\frac{\psi}{2}) - cos(\frac{\phi}{2})sin(\frac{\theta}{2})sin(\frac{\psi}{2}), \\ cos(\frac{\phi}{2})sin(\frac{\theta}{2})cos(\frac{\psi}{2}) + sin(\frac{\phi}{2})cos(\frac{\theta}{2})sin(\frac{\psi}{2}), \\ cos(\frac{\phi}{2})cos(\frac{\theta}{2})sin(\frac{\psi}{2}) - sin(\frac{\phi}{2})sin(\frac{\theta}{2})cos(\frac{\psi}{2}) \end{pmatrix} \tag{3.10}$$

From there we can generate the rotation matrix using the quaternion element, q = (a, b, c, d) (Eq.3.11).

$$R = \begin{bmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{bmatrix} \tag{3.11}$$

And calculate the movement vector with the correct orientation using the matrix multiplication shown in Eq.3.12.

$$\vec{m} = R * \vec{v_n} \tag{3.12}$$

III Update Position

Updating the position is just the matter of adding the movement vector, $\vec{m}$, to the current position, $\vec{p}_c$ as shown in Eq.3.13:

$$\vec{NewPosition} = \vec{p}_c + \vec{m} \tag{3.13}$$

3.4.2 Interactions

Design of interaction techniques and user interfaces for VR must be done with extreme care. Typical designs are fundamentally tweaked to 2D interactions (with a mouse and keyboard) but VR is fundamentally 3D. The type of interaction technique will depend on the task to be performed. In our case, we are interested in use the best interaction techniques to assist in visualization of scientific data and the steering of simulations.

3.4.2.1 Selection

Given the expected size of scenes, for selections, we will just consider options that solve the problem of objects not within reach. Between the two techniques described in the state of art that meet that criteria (Hyper-Natural Interaction, and Magic Interaction) we believed that the fact of having no reach limitations, being the *de facto* standard 3D selection technique in immersive environments [76] and having good performance results [106] were enough pros to make "Magic Interaction" the way to go.

In magic interaction (alias ray casting or virtual laser pointer), a light ray (or laser beam) is cast from user's hands (or more concretely from the VR controllers) and the intersections with objects in the scene are evaluated. When the desired object is under hover (which should be made clear through visual feedback), the user should press a button to confirm the selection (this action should, as well, be made clear through some kind of feedback, typically visual).

We define a ray to be a direction vector $\vec{u}_r$ with an origination point p_r (Fig.3.9). Distance along the ray is measured by parameter t. Given this definition, we can map all the points x lying on a ray by formulating the ray equation 3.14.

$$x(t) = p_r + t * u_r, \quad t \geq 0 \tag{3.14}$$

To determine if a ray hits an object we look for any points in the surface of an object that satisfy equation 3.14. For selection reasons, if more than one object satisfies the condition, the closest to the viewpoint (lower t) is the one selected.

3.4.2.2 Manipulation

There are numerous ways to manipulate objects, some of the most important ones, and the ones we are going to describe are: rotation and scaling. Once an object (or set of objects) has

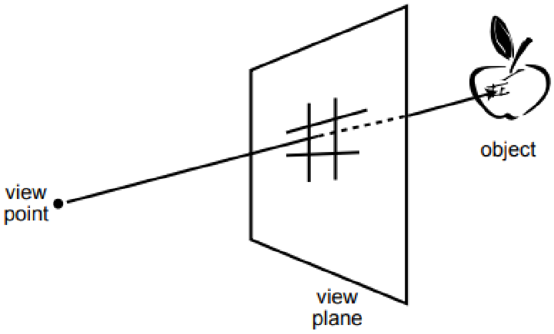

Figure 3.8: Ray "shooting" example

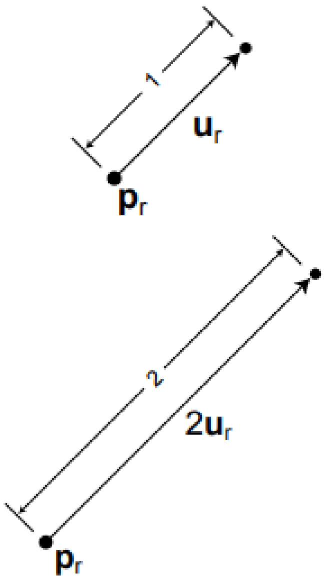

Figure 3.9: Ray example

been selected, the user can now manipulate it. We decided to support both the stick method, which applies the same transformations the user does in the VR controller of his/her dominant hand into the object, and the popping technique which allows the object to be brought to within reach for manipulation and after finishing it's placed back in its original position. The popping technique has the potential to allow manipulations to be even more realist and natural if the headset has hand tracking capabilities and has the obvious advantage of having the object to manipulate fully visible.

I The stick method

Let the beginning and ending of manipulation be t_0 and t_1, ł_h and r_h represent the left hand and right hand respectively, $\text{ł}_{\vec{hPos}}$ and $r_{\vec{hPos}}$ the position vectors of both hands and $\text{ł}_{\vec{hRot}}$ and $r_{\vec{hRot}}$ the orientation vectors. Both the implemented interactions will be further explained and assume that the manipulation beginning time t_0 has already started.

Rotation

The rotation of the object follows the same rotation of the hand (VR controller) that started the manipulation (for the purpose of this explanation lets assume it was the right hand, r_h). We start by calculating the rotation that happened in the controller between ticks (time instants), $\vec{\Delta r}$, Eq.3.15,

$$\vec{\Delta r} = (r_{\vec{hRot}}.x[t] - r_{\vec{hRot}}.x[t-1], r_{\vec{hRot}}.y[t] - r_{\vec{hRot}}.y[t-1], r_{\vec{hRot}}.z[t]) \quad (3.15)$$

where t is the current tick and $t-1$ the previous one. Note that, for practical reasons, the rotation over the z axis was not considered.

The rotation of the object, $\vec{o_r}$, is then calculated as stated in equation 3.16.

$$\vec{o_r}.x = r\vec{_{hRot}}.x[t_0] + \vec{\Delta r}.x, \quad \vec{o_r}.y = r\vec{_{hRot}}.y[t_0] + \vec{\Delta r}.y, \quad \vec{o_r}.z = \vec{o_r}.z \tag{3.16}$$

Scaling

The scaling is performed by a gesture which requires both hands (VR controllers). When the hands start to move in opposite directions parallel to the ground (Eq.3.17) we identify that the object should be scaled up. The opposite movement (Eq.3.18) identifies that the object should be scaled down.

$$r\vec{_{hPos}}.x[t] - r\vec{_{hPos}}.x[t-1] > 0 \quad \wedge \quad l\vec{_{hPos}}.x[t] - r\vec{_{hPos}}.x[l-1] < 0 \tag{3.17}$$

$$r\vec{_{hPos}}.x[t] - r\vec{_{hPos}}.x[t-1] < 0 \quad \wedge \quad l\vec{_{hPos}}.x[t] - r\vec{_{hPos}}.x[l-1] > 0 \tag{3.18}$$

where t is the current instant and $t-1$ the previous one.

Whenever one of those equations is true, we scale the object accordingly. Let $v = (v_x, v_y, v_z)$ represent the vector that contains the correct (according to the situation) predefined scale values and $p = (p_x, p_y, p_z)$ a point of the object to scale. Equation 3.19 shows how to acquire the new point position. The process should be applied to all points of the object.

$$S_v p = \begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} v_x p_x \\ v_y p_y \\ v_z p_z \end{bmatrix} \tag{3.19}$$

where S_v is the scaling matrix.

II The popping technique

The popping technique is very simple, it starts by saving the initial object position, $\vec{o_{p0}} = \vec{o_p}[t_0]$, gets the user position, $\vec{u_p}$ and adds to it a comfortable z delta, Δz, so that the object gets in front of the user and not exactly where the user is (Eq.3.20).

$$\vec{o_p} = (\vec{u_p}.x, \vec{u_p}.y, \vec{u_p}.z + \Delta z) \tag{3.20}$$

Note that all the position vectors should be relative to the world referential and not to local referential. If the later happens make sure to convert them as explained below:

Let $LR = (r_1, r_2, r_3)$ be the local reference matrix, where $r_1 = (a_1, b_1, c_1)$, $r_2 = (a_2, b_2, c_2)$ and $r_3 = (a_3, b_3, c_3)$, the world position is given by equation 3.21.

$$\begin{bmatrix} x_{world} \\ y_{world} \\ z_{world} \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x_{local} \\ y_{local} \\ z_{local} \end{bmatrix} \tag{3.21}$$

3.5 Summary

Throughout this chapter, the different VR techniques for navigation, selection and manipulation were discussed. Real physical rotation served as basis for the navigation supported by joysticks. The use of a raycaster together with the stick method and the popping strategy were used to select and manipulate objects respectively.

The fundamentals of biophysical computing tools and its realization, were explained to the level of detail that was considered appropriate for understanding its methodology and its role in this dissertation.

The following chapter will make use of the concepts addressed here but from a practical point of view, with its application in the prototype.

Chapter 4

Implementation

This chapter is dedicated to the practical description of the visualization prototype (section 4.1), corresponding architecture (section 4.2) and implementation details, such as, the technologies used and why (section 4.3), the interactions made available and how they were achieved (section 4.5), the scenes accessible to the user and what they represent (section 4.7) and explanations on the process of running simulations (section 4.6) and on the sequence of steps necessary to go from the model definition to viewable data (section 4.4).

4.1 Prototype

The visualization prototype implemented aims to demonstrate the main functionalities of the platform.

When running the prototype, the user can select one of 3 scenes (Fig.4.1, Fig.4.2, Fig.4.3). The available scenes correspond to different models of different neuroscientific data. Users are able to interact with the scene through the use of VR controllers. Although quite simplistic, this prototype, in addition to exploring the technical platform, introduces several interesting concepts such as free navigation and possibility to interact closely with sub-parts of the visualized model.

At the same time, the possibility of simulating a small experiment is provided, with the injection of potential on the visualized cells (Fig.4.4, Fig.4.5). Additionally, in the case of the CA1 Pyramidal Cell scene [4.7.2], there's also an option to show visual groups (Fig.4.6). Visual groups allow the users to visualize an entity, colouring its different elements according to a set of properties related to the entity itself. A visual group can, for instance, allow the user to see the different cell regions of a neuron or the distribution of one type of ion channel.

The further exploration of the components showcased in the prototype, can lead to innovative and highly efficient experiences in the visualization of scientific data.

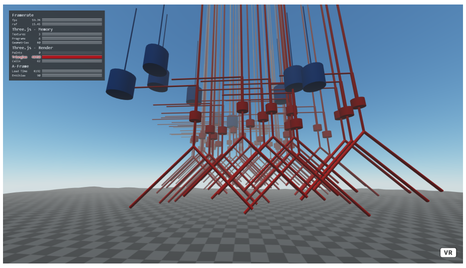

Figure 4.1: Auditory Cortex scene

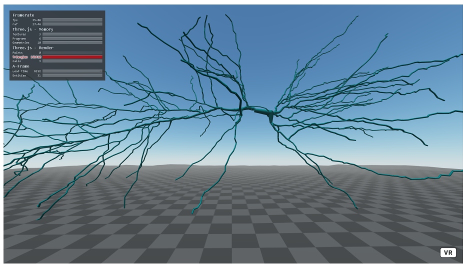

Figure 4.2: CA1 Cell scene

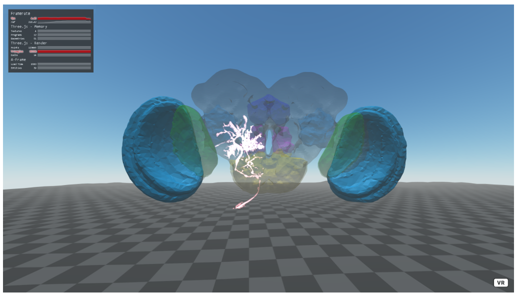

Figure 4.3: Virtual Fly Brain scene

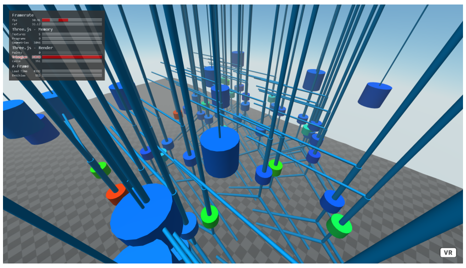

Figure 4.4: Auditory Cortex simulation

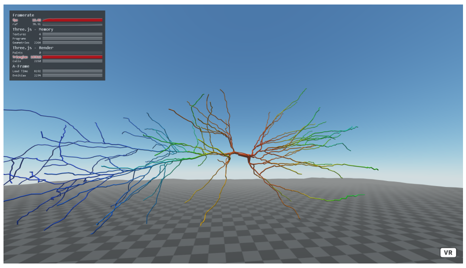

Figure 4.5: CA1 Cell simulation

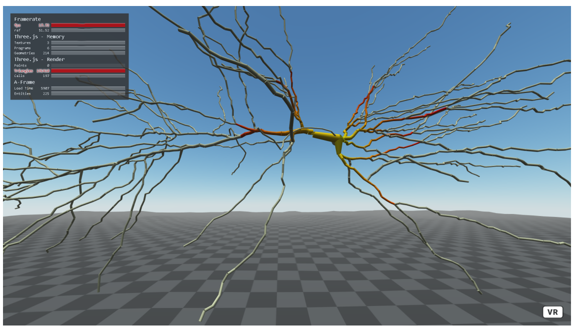

Figure 4.6: CA1 visual group that highlights potassium channels

4.2 Architecture

Figure 4.7 presents a high-level view of the implemented solution. In the diagram, client applications (representative of user VR headsets) access via network (e.g. Internet) to the server application. This communication is made using the standard web communication protocol (http) with an additional layer of security required by the WebXR protocol (ssl / tls). As the role of the server in the prototype is just to serve static scientific data, it was not necessary to use a database. However, in a real application this would possibly have to be included in the diagram to deal with more realistic data manipulations.

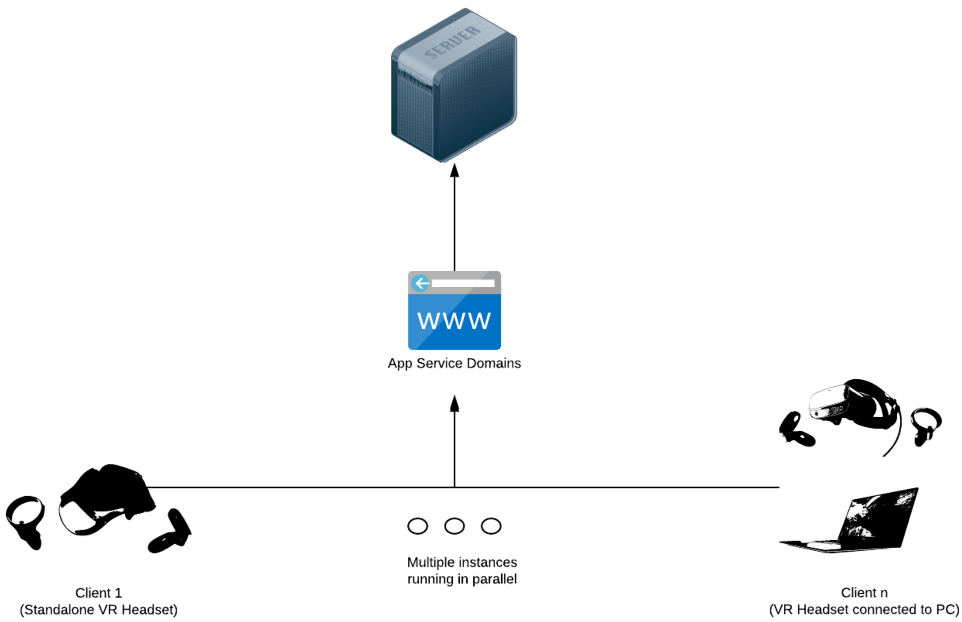

Figure 4.7: A high-level view of the implementation architecture used

4.3 Technologies

The developed prototype combines multiple technologies taking into account the different needs of the project: scientific model interpreter, VR scene creator and interface provider.

For the scientific model interpreter, as stated in section 3.3.1, we decided to use Geppetto because, by supporting simulator-independent markup languages, such as NeuroML or LEMS, we managed to cover a wider range of neuroscientific models. Something that wouldn't be possible, without the extra work of adding support to it, with other of the tools studied.

Before deciding the library to help us create the VR experience, we should first remember that our prototype should be able to generate custom 3D objects (due to the specificity of the data to display), work as a web application (which was an agreed requirement) and that just recently the WebXR protocol was implemented (and the WebVR is going to be discontinued).

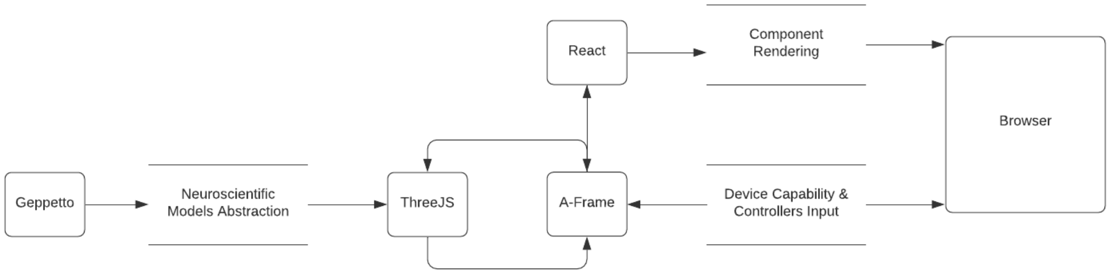

Figure 4.8: Technology stack diagram

Given this information and the set of options available (A-Frame, React360, or using a game engine), we initially removed using a game engine of the equation due to the ones we considered (Unity and Unreal Engine) had limited or no support (respectively) to WebXR. After this first elimination, the decision was simple as A-Frame is the only mature framework that allowed the creation of custom 3D objects by relying in ThreeJS [15] (a javascript library created to generate 3D objects).

A-Frame provides a number of primitives out of the box. Developers can create custom primitives as well by registering them through JavaScript. It also contains an asset management system, which holds all assets (textures, sounds or videos) at one place and improves performance by preloading and caching them. Typically those components interact with each other throw the usual events and event listeners interface. The community around the framework is very active and therefore numerous open-source components are available to use. Unfortunately, due to the transition times at which this prototype was implemented, a large amount of them were incompatible with the latest WebXR protocol and thereafter with the latest A-Frame version (at the time of writing, v.1.0.4). This fact alone, had a big impact in what could be or couldn't be done in the time frame of this thesis, as most of the interactions had to be written from scratch (which was not expected at proposal time). Although typically used for standalone projects by being HTML based, A-Frame can be plugged into any framework such as React.

This bring us to the last piece of the puzzle, the interface provider, here the best decision from a performance point of view would be to use no extra interface provider as A-Frame can deal with that on its own. However, due to intersections with the company strategies and standards we decided to use React [4].

Figure 4.8 shows the technologies interconnection. In the diagram we can see the Geppetto framework being used to define (declaratively) the 3D structures of the neuroscientific models - primitives and respective positions and/or external resources to load. This information is filtered and parsed so that it can be used by the ThreeJS library to create the 3D meshes accordingly. A-Frame interacts with ThreeJS to produce the VR experience and updates all the meshes in the scene accordingly to the browser information regarding the user's interactions. React is used to create the user interface.

4.4 The visualization pipeline

As described in 2.2.3, a visualization process typically follows an iterative sequence of steps to provide the best experience to the users. That pipeline, represented in Fig.4.9, is now explained in detail.

The block *Geppetto instances* represents the data that contains all the (most recent) information regarding the simulation/model to display. That information is provided to the *Canvas* component that is responsible to create/update the scene. To do that it delegates the filtering and mapping of the data from regular data to renderable primitives to the *GeppettoThree* module. In this module, the conversion process starts by traversing all the instances (in *traverseInstances* function) and looks for the ones with visual capabilities - instances from the type Visual (this type is defined in the model) containing methods specific for Visual variables (e.g *getPosition()*, *getVisualType()*) (in *checkVisualInstance* function). Instances with visual capabilities are structured hierarchically and can contain information regarding multiple meshes, so we iterate over that structure (in *walkVisTreeGen3DObjs* function) and create each one of the meshes accordingly to the class - cylinder, sphere, obj, collada, particles - they belong to (in function *create3DObjectFromInstance*). If the complexity of the scene is over the threshold defined, cylinders can be simplified into lines. If not specified otherwise, we try to merge all the meshes created into one single mesh (in function *generate3DObjects*) so that we can decrease the scene complexity. The mesh created in this process, with both geometry and material defined, is now saved in a dictionary, with the position in the scene specified, along with other metadata (in function *init3DObject*). The *Canvas* component renders the scene based on the information present in the dictionary.

4.5 User Interactions

The system surrounding the use of this prototype includes the application, the headset and the VR controllers (Fig.4.10).

The set of interactions available contains: navigation (it is obtained by a combination of the user changing the orientation of the headset and using the thumbstick in the VR controller), selection (the user must point the VR controller the element he/she wants and click on a button in the controller to select it), both rotation and scaling of objects (these actions are performed by keeping a button in the controller pressed, followed by a determined gesture - hand rotation or arms movement respectively) and an custom interaction denominated 'Bring Closer' which refers to the ability of bringing an object closer to the user (this can be achieved by clicking in a button on the VR controller). Figure 4.11 summarizes these interactions.

A short video[1] describing side by side how to perform the interactions and what is the effect on the scene was created and is available publicly.

[1] `https://youtu.be/LN0IB1WC_0k`

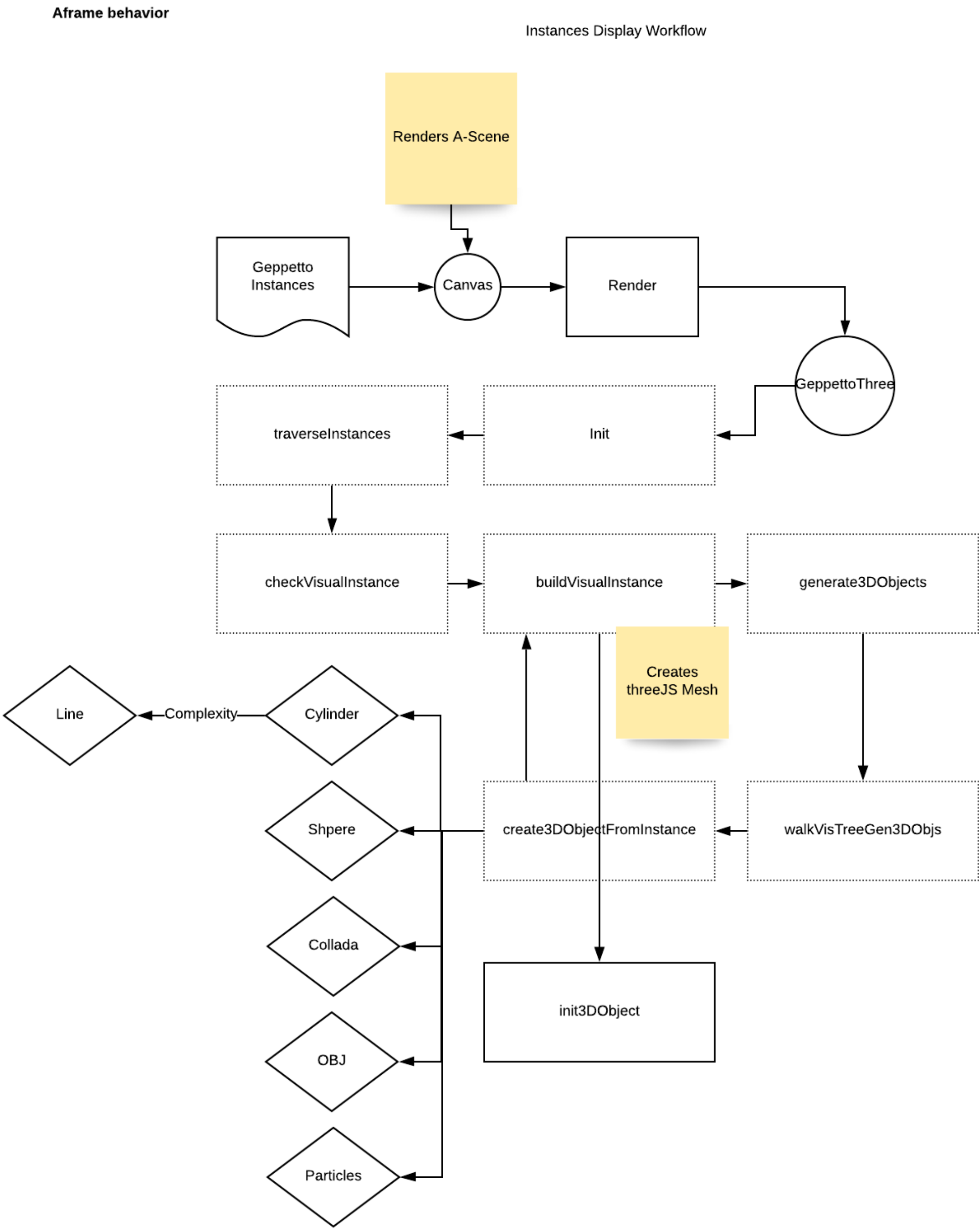

Figure 4.9: Canvas workflow diagram

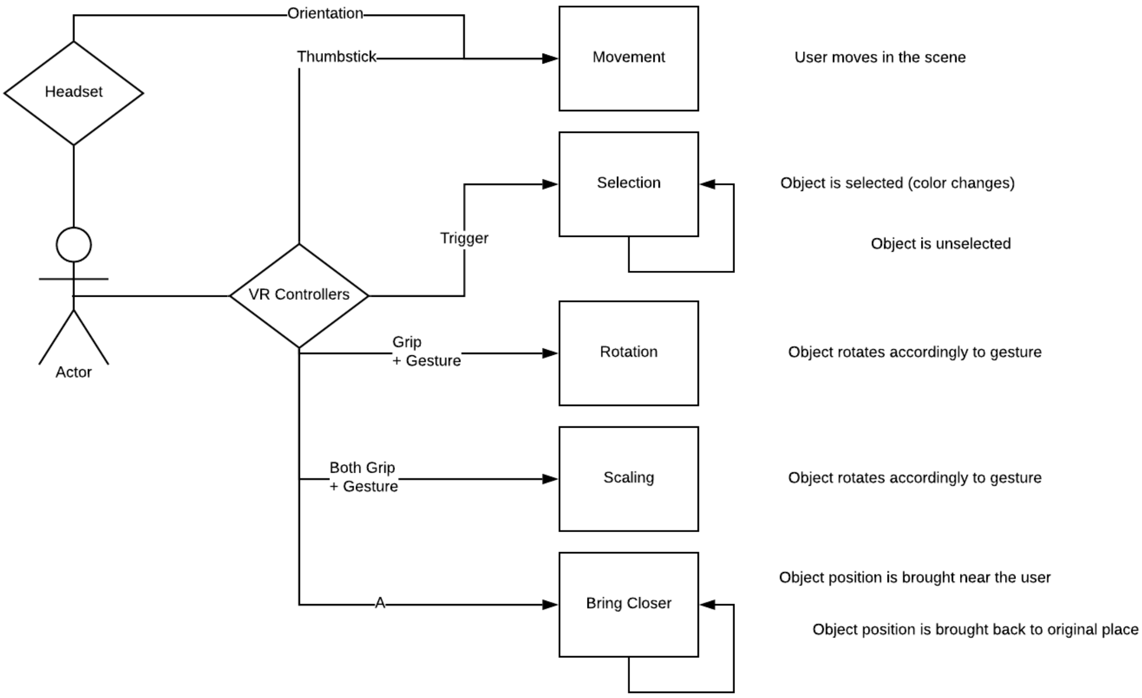

Figure 4.11: User interactions diagram

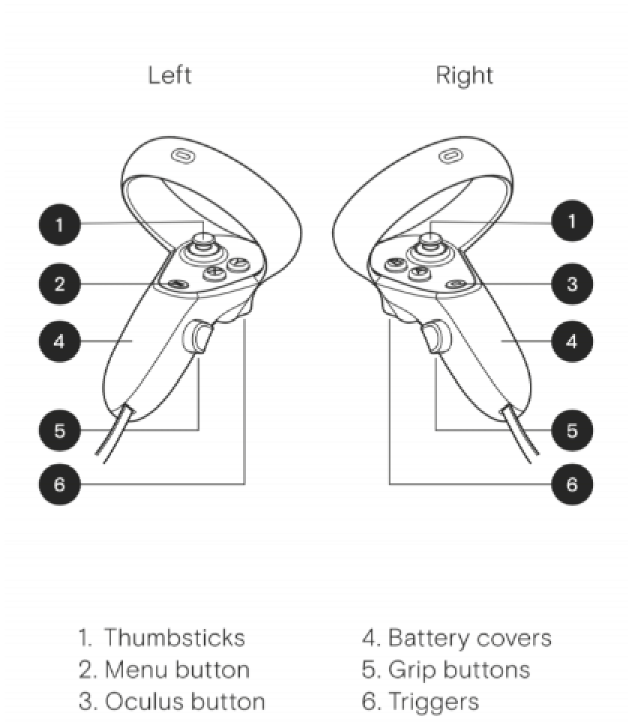

Figure 4.10: Oculus controllers schema

4.5.1 Navigation

The movement works as explained in section 3.4.1. Whenever the user moves the thumbstick (Fig.4.10 1) (event 'thumbstickmoved') the orientation of the headset and the thumbstick event information (x, y values referring to the latest change in the thumbstick position) are considered to calculate the new position. This behavior happens under the tick function (handler which is be called on every frame) and only stops when the movement in the thumbstick stops (event 'thumbsticktouchend').

4.5.2 Selection

The selection technique was explained in section 3.4.2.1. A raycaster is constantly projecting rays from the VR controllers and detects the intersection with the objects in the scene that are intersectable (and only those). The latter can be seen as an optimization that allows the raycaster to not have to calculate the possible collisions with all the objects in the scene, but only the ones of a certain class. When a intersectable object is intersected by the raycast laser (event 'mouseenter') the color of the object changes providing visual feedback to the user. If the raycast laser no longer intersects the object (event 'mouseleave') its color gets back to the original. If the user is happy with the object 'hovered' he can then opt to select it by clicking in the trigger button (Fig 4.10 6) (event 'triggerdown') of the VR controller. Once again this provides visual feedback to the user by changing the color so that he/she can have a confirmation for the action. If the user 'clicks' in a object that was previously selected, the action will unselect it.

4.5.3 Rotation

The rotation strategy was described in section 3.4.2.2. The idea here is to take advantage of the (at least) three degrees of freedom of the VR controllers, and whenever the user presses the grip button (Fig.4.10 5)(event 'gripdown'), we retrieve its orientation and use it to rotate the objects selected in the scene (or the full model if none is selected). The rotation stops when the grip button is released (event 'gripup').

4.5.4 Scaling

Scaling, which was explained in 3.4.2.2, is very similar to the rotation in terms of procedure. It can be said that it is a little bit more complex as it evolves the two controllers and not just one. Whenever the grip buttons of both VR controllers are pressed, the application will listen for the horizontal movement of the user arms. If the arms are spreading the objects selected (or the model if none was selected) will scale up. If the opposite arms movement is detected, the objects will scale down. The scaling finishes when at least one of the grip buttons is released.

4.5.5 Bring Closer

The interaction nicknamed 'bring closer' is no more than the implementation of the popping technique shown in 3.4.2.2. When the user presses the A button (in the right controller) (event 'abuttondown') the position of the object is updated to the front of the user. This happens by collecting the user position in the scene, add to it a predefined delta in the sense the user's facing, into the z coordinate (depth) and apply that value to the object. A second click in the A button will move the object back to its original place.

4.6 Simulations

In our prototype we allow the running of a simple simulation (in two of our scenes, CA1 Pyramidal Cell and Auditory Cortex). In order to restrict the prototype to the visualization task, the simulations were run in parallel and their output recorded. The format of the output consists in a *outputMapping* file which tells us which variables were recorded, and multiple *results* files containing the values recorded during the full time of the experience. The application developed had to read from those files and structure the data (into a nested map of the type: Timestamp$\rightarrow$ Variable$\rightarrow$Value). The values, in our case, were the electric potential of the different parts of the neurons. To show the simulation running, we iterate over the structured data over time and associate a colour to the different values of the electric potential (following the color scheme of Fig.4.12).

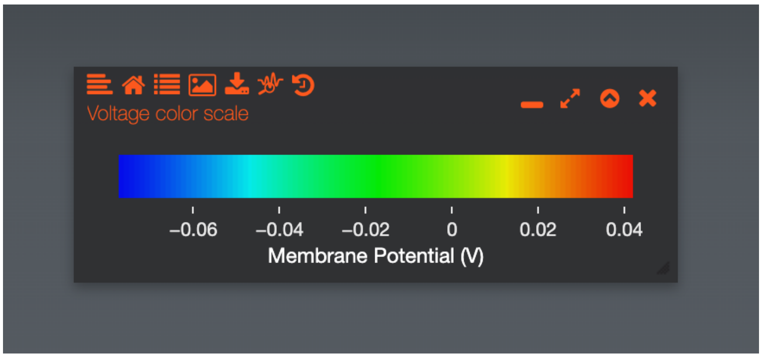

Figure 4.12: Voltage color scale

4.7 Scenes

To showcase the potential of the prototype, different scenes with different levels of analysis and detail were made available. Auditory Cortex is the default scene which shows a small network of cells. CA1 Pyramidal Cell and VFB are the other two scenes available to explore. The first shows a morphology of a cell in detail, while the second represents a fly's brain model for high-level analysis.

4.7.1 Auditory Cortex

A small scale network model with 60 cells based on Dave Beeman's Auditory Cortex model [5]. Contains two populations: 12 cells of type bask - a simplified basket cell model with 2 compartments; 48 cells of type pyr_4_sym - a simplified pyramidal cell model with 9 compartments. Basket cells are inhibitor interneurons of the brain, while pyramidal cells are the primary excitation units of the mammalian prefrontal cortex. Figures 4.1 and 4.4 shows the look of the scene in the prototype.

4.7.2 CA1 Pyramidal Cell

A detailed cell model from CA1 pyramidal cells based on Migliore et al [63]. This scene contains 2243 segments to represent the full morphology of only one cell. Multiple color mappings can be applied in this cell to highlight different aspects of the cell (show different cell regions, show different ion channels). Figures 4.2, 4.5 and 4.6 shows the look of the scene in the prototype.

4.7.3 Virtual Fly Brain

The Virtual Fly Brain (VFB) scene is based on the Virtual Fly Brain project [65]. It is an example of a higher level of analysis as it allows the users to explore slices of the adult fly brain. The modulation is based on the article by Arnim Jenett et al article [47]. In our prototype, only a few of the slices were made available, namely: the deutocerebrum of the adult brain, the adult gnathal ganglion (region of the adult brain beneath the esophagus), the fan-shaped body (largest synaptic neuropil domain of the adult central complex), the posterior ventrolateral protocerebrun (glomerular, bilaterally paired synaptic neuropil) and the medulla (second optic neuropil, sandwiched between lamina and the lobula complex) - which is not seen in Fig.4.13 because it is encapsulated by the deutocerebrum. One particular characteristic of this scene is the presence of "particles" which are a large cluster of points with a texture applied. This memory intensive representation is used in our scene to represent the antennal mechanosensory and motor center AMMC Di7 neuron. Figures 4.3, shows the look of the scene in the prototype.

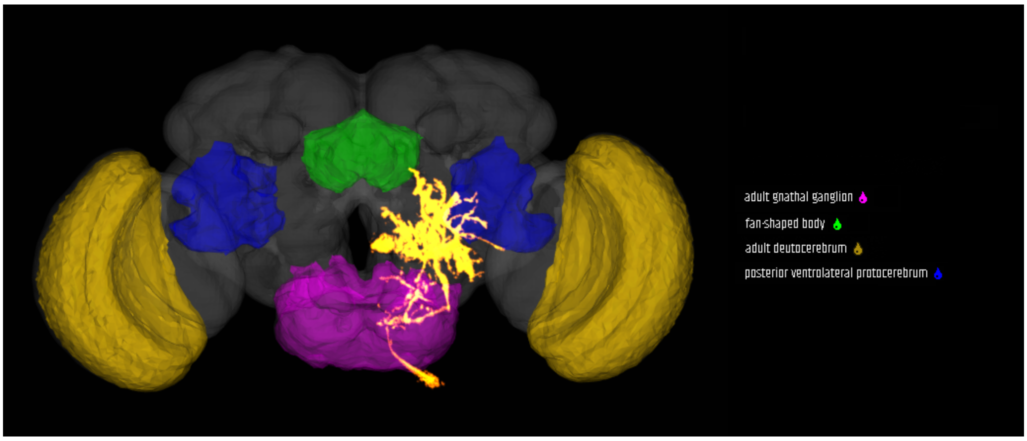

Figure 4.13: VFB slices with color mapping

4.8 Summary

In this chapter we described the practical portion of the visualization prototype developed, corresponding architecture and implementation details. This details contain information such as: the reasons why A-Frame was selected as the library to use to provide the VR experience, its relationship with both ThreeJS and React and the role of Geppetto in this technology stack; the steps needed to convert the data described in the model into renderable objects; the interactions available and the events and logic behind them; as well as some notes explaining the simulation process and the scenes accessible to the user.

The next chapter will be dedicated to evaluate the quality of the solution proposed in this dissertation by performing 3 different studies.

Chapter 5

Evaluation and Discussion

This section presents and discusses the 3 studies carried out to evaluate the quality of the solution proposed in this dissertation: one to evaluate the acceptance of VR into computational biophysics; and two others to get feedback on the prototype developed, one practical and one theoretical.

5.1 Acceptance Evaluation

Understanding the acceptance of VR into computational biophysics was the first thing we wanted to evaluate. To do so, we adopted the Technology Acceptance Model (TAM) [23] (Fig.5.1) to predict and explain the intention of scientists, developers and other relevant personalities of the field of computational biophysics to use VR for biophysical scientific research. At the end of the evaluation we hope to be able to identify which characteristics of VR are relevant for Computational Biophysics and speculate on the why.

5.1.1 Technology Acceptance Model

TAM is an information systems theory that models how users come to accept and use a technology. Davis [23] hypothesizes the behavioral intention (BI) determines the technology usage. He also states that BI is influenced by the attitude (A), which in turn, is influenced by the perceived usefulness (U) - the degree to which a person believes that using a particular system would enhance his or her job performance - and perceive ease of use (E) - the degree to which a person believes that using a particular system would be free from effort.

5.1.2 Participants

A questionnaire was created [Appendix A.1] and shared with MetaCell's members and clients. We obtained 15 samples. Table 5.1 summarizes the socioeconomic characteristics of the respondents. 80% of the respondents were male, and 20% were female. The majority of the respondents were aged between 31 and 50 (86.7%), followed by those aged between 18 and 30 years old

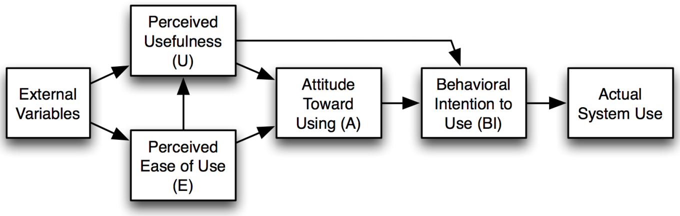

Figure 5.1: The Technology Acceptance Mode

(13.3%). Of the 15 respondents, 40% had less than 10 VR experiences, followed by those with no experience whatsoever and between 10 an 100 experiences (26.7%) and 6.7% had over 100 experiences. 46.7% of the participants had over 100 experiences visualizing biophysical data, on the opposite side 33.3% had no experience at all, while 13.3% had between 10 and 100 experiences and 6.7% less than 10. All the participants were high educated (college level) and more than half (66.7%) had as occupation something in the field of biophysics or/and software development.

5.1.3 Results

We divided our results analysis in two phases. Phase one corresponds to an exploratory factor analysis which aims to validate the form. Phase two corresponds to confirmatory factor analysis with the goal to estimate the relationship between the variables in study. Each stage will be specified in detail below:

5.1.3.1 Exploratory factor analysis

We started by performing an Exploratory Factor Analysis (EFA) to check the factor structure of the empirical data and determine the degree of correlation between each factor and each observed variable. The EFA was performed in an iterative way so we could conclude what the best algorithm to use and what questions from the questionnaire should be kept. Due to overall better results according to the EFA best practices we opted to use Partial Least Squares (PLS) algorithm and remove questions E1 and B4. With this changes we were able to keep the composite reliability (CR) above 0.6 in all factors which indicates item reliability [3] and all factor loadings (λ) above 0.5 which indicates convergent validity [38]. Discriminant validity is demonstrated in table 5.2: the square root of the average variance extracted (AVE) value for each factor is greater than the correlation between the different factors [30]. Table 5.3 shows the results of this step.

5.1.3.2 Confirmatory factor analysis

The main goal of confirmatory factor analysis is to test if the model fits the actual data. To do so we started by performing a path analysis to estimate the relationship between the variables in the proposed model and determine which variables are cared for by the users. Figure 5.2 shows the result of that step. Each node represents one of the factors considered in the original TAM

Attributes	Sub-Groups	Frequency(N=15)	Percentage (%)
Gender	Male	12	80
	Female	3	20
Age Group	18-30	2	13.3
	31-50	13	86.7
Education Level	Bachelor	7	46.7
	Graduate	5	33.3
	Doctoral	3	20
Occupation	Bioinformatic Researcher	1	6.7
	Biologist	1	6.7
	Software Engineer/Developer (for scientific related projects)	8	53.3
	Other	5	33.3
VR Experience	None	4	26.7
	Less than 10 experiences	6	40
	10-100 experiences	4	26.7
	Over 100 experiences	1	6.7
Experience of visualizing biophysical data	None	5	33.3
	Less than 10 experiences	1	6.7
	10-100 experiences	2	13.3
	Over 100 experiences	7	46.7

Table 5.1: Demographic profile of the respondents.

	A	BI	E	PU
A	**1**			
BI	0,616	**0,774**		
E	0,665	0,0404	**0,895**	
PU	0,012	-0,308	-0,227	**0,793**

Table 5.2: Inter-construct correlations as discriminant validity (square root of AVE in diagonals).

Constructs	Indicators	Mean	SD	λ	AVE	CR
Perceived Usefulness	PU1	3.8	0,748	0,905	0,629	0,866
	PU2	3.8	0,833	0,974		
	PU3	3.86666666666667	0,957	0,635		
	PU4	4.2	0,748	0,587		
Perceived Ease of Use	~~E1~~	3,866666667	1,024		0,801	0,923
	E2	4	1,033	0,839		
	E3	4	1,095	0,953		
	E4	3,8	1,116	0,889		
Attitude Towards Using	A1	3,8	1,108	1	1	1
Behavioral Intention	BI1	4,333333333	0,596	0,739	0,599	0,817
	BI2	3,066666667	1,123	0,816		
	BI3	2,733333333	1,236	0,765		
	~~B4~~	3,666666667	0,869			

Table 5.3: Validity and reliability of the measurement model.

(Perceived usefulness (PU), Perceived ease-of-use (E), attitude towards using (A) and Behavioral Intention (BI)). Each one of the yellow rectangles (PU1, E2, BI1...) represent each question of the questionnaire that was considered for this study. The directed edges between nodes and questions represent the factor loadings (correlation coefficients between observed variables and latent common factors). The directed edges between nodes represent the effect of the node where the edge leaves into the node where the edge arrives. The values in each node are the average variance extracted of the factor.

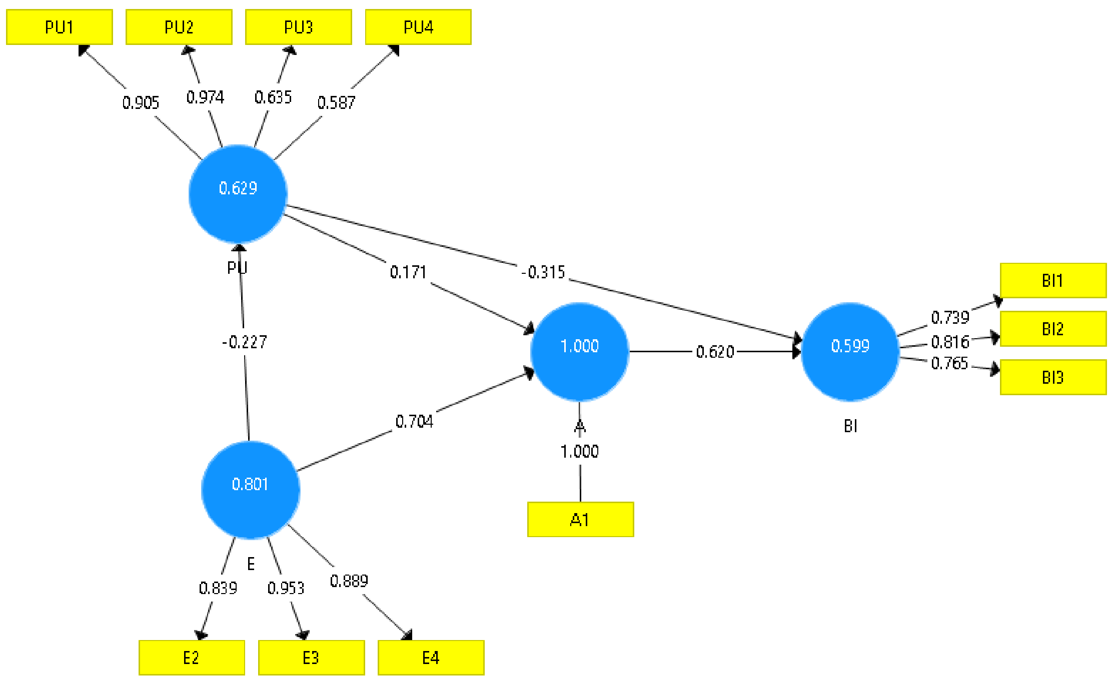

Figure 5.2: Path verification

The next step of the confirmatory factor analysis is to calculate the model fit indexes to determine whether the model fit is good. The results of this task are not encouraging, since the recommend criteria for all the index studied are out of the recommended space. A possible justification for this unfitness is the low number of participants [50]. Table 5.4 shows how the model performs compared with the recommended criteria. Standardized Root Mean Square Residual (SRMR) is defined as the standardized difference between the observed correlation and the predicted correlation, it's recommended a that its value should be less than 0,08. The chi-square test is generally a reasonable measure of fit, in those circumstances the recommend value is to be lower than 5.0, however with small sample sizes, this measure is not very indicative. The normed fit index (NFI), suffers from the same problem, as it analyses the discrepancy between the chi-squared value of the hypothesized model and the chi-squared value of the null model.

	Results	Recommended Criteria
SRMR	0,155	< 0,08
Chi-Square	63,738	< 5.0
NFI	0,522	> 0.85 (close to 1)

Table 5.4: Model fit indexes and recommended criteria.

5.1.4 Discussion

This study developed a theoretical framework based on the TAM model. The original model expects a positive influence of perceived ease of use over perceived usefulness which is not supported in our data. The same goes to the influence of perceived usefulness on behavioral intention. These inconsistencies can be caused by the very dependent behaviour of perceived usefulness and perceived ease on the sample, study context, and technology under consideration. With that in mind, analysing the latter conclusion suggests that there should be more effort put on individuals' perception of usefulness of VR for computational biophysics. This could be done by (a) improving user perception of the determinants of perceived usefulness such as information quality and/or (b) organization of campaigns that aim to promote the benefits of VR for computational biophysics [27]. The positive effect of perceived ease of use on attitude towards use and the effect of attitude towards use on behaviour intentional are consistent with the original model. Considering the sensation of 'real' provided by VR, the rapid development of ease of use has become a basic characteristic of VR applications. The smaller influence of "perceived usefulness" over "attitudes towards using" compared with the influence of "ease of use" on "attitudes towards using" also, suggests that users value more the ease of use of the VR technology over its usefulness in the computational biophysics field.

This proposed TAM model can have some practical and theoretical implications for researchers and engineers to develop popular VR services for computational biophysics. This study provided some analysis of the acceptance of VR for scientific visualization and research with a small sample of engineers and researchers of the field. This information can potentially be applied in current and future computational biophysics tools and contribute to the evolving of the field.

5.2 Controlled Experiment

To assess the quality of the proposed solution as a whole, we conducted a controlled experiment. We can summarize the scope of our experiment using the framework proposed by Wohlin and extended by Merino as follows:

> "Analyze [3D visualizations] in a [web-based application environment] to support the [visualization and interaction tasks] using [neuroscience data] displayed on a [computer screen or VR headset] for the purpose of [comparing both visualization and interaction techniques] with respect to [effectiveness and usability] from the point of view of [the users] in the context of [biophysical visualization tools]." [62]

In other words, we use this framework to specify an experiment which aims to analyze if there are any advantages for the users of a biophysical visualization tool, in terms of effectiveness and usability, regarding visualization and interaction of and with 3D structures (related with neuroscience), in using a VR headset over a computer screen.

The following subsections formulate experiment that was carried out to evaluate the impact of the visualization change.

5.2.1 Design

The experiment will be split in two phases, so that the participants will have more context and find it easier to judge which version they prefer (the same person tests all the conditions - within-subjects approach):

- Phase A - Using a standard computer display the user will perform the task. The result of this phase will be used as baseline.

- Phase B - Using VR headsets the user will perform the task. The result of this phase will be compared with the baseline.

The task to be performed relies in the same subset of actions in both phases of the experiment. The subset of actions contain both navigation and selection procedures.

Phase A will be able to use the keyboard to move and the mouse to interact with the scene.

Phase B will be able to use the movement of the head to orientate themselves in the scene and VR controllers to navigate and interact with it.

Both phases should be completed in less than 15 minutes to avoid stress. The order in which the phases were taken was randomized between the participants (participant 1 does phase A 1st, participant 2 does phase B 1st, and so on...).

5.2.2 Hypothesis

The following parameters were defined to be under evaluation:

- The time to complete the tasks.

- The correctness of the tasks.

- The difficulty perceived by the participants.

The experiment will support the research issue if it fulfils at least one of the following hypothesis:

H1 - The time to complete the tasks on Phase B is not higher than on Phase A.

H2 - The correctness of the tasks on Phase B is not smaller than on Phase A.

H3 - The difficulty perceived by the participants regarding the tasks on Phase B is not higher than on Phase A.

Attributes	Sub-Groups	Frequency(N=8)	Percentage (%)
Gender	Male	5	62.5
	Female	3	37.5
Age Group	Under 18	1	12.5
	18-30	7	87.5
Education Level	High School	3	37.5
	Bachelor	4	50
	Graduate	1	12.5
Occupation	Biologist	1	12.5
	Software Engineer/Developer (for scientific related projects)	3	37.5
	Other	4	50
VR Experience	None	4	50
	Less than 10 experiences	2	25
	10-100 experiences	2	25
Experience of visualizing biophysical data	None	6	75
	Less than 10 experiences	1	12.5
	10-100 experiences	1	12.5

Table 5.5: Participants demographic and previous expertise information

5.2.3 Participants

A group of 8 participants was selected without any requirement regarding VR or scientific visualization expertise, opposite to what was initially desired. All participants completed questionnaires providing demographic information and level of expertise on VR and scientific data visualization. The information resultant from the questionnaires is shown in the table 5.5.

62.5% of the respondents were male, and 37.5% were female. The majority of the respondents were aged between 18 and 30 (87.5%), followed by those aged under 18 (12.5%). Of the 8 respondents, 50% had no experience whatsoever with VR, followed by those with less than 10 VR experiences and between 10 an 100 experiences (25%) each. 75% of the participants had no previous experience visualizing, biophysical data, while 25% had less than 10 experiences or between 10 and 100 experiences (12.5% each). The education level and occupation were not considered for this test.

5.2.4 Procedure

The software applications and hardware devices used in the experiments were the same for all the participants. The scene used for the experiment was the Auditory Cortex.

The task was to search, locate and select an object, previously flagged with a "X". The object flagged cannot be easily distinguished from the rest of the objects without a careful exploration of the scene. The users are allowed to use any of the object manipulations available (rotation, scaling, bring closer). In phase A, the participants used a high performance laptop and interacted with the visualization tool with mouse and keyboard. In phase B, the users used an Oculus Quest

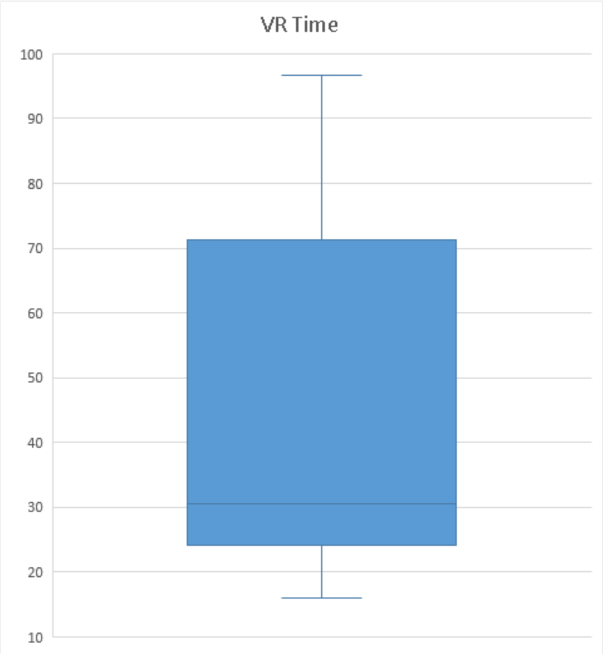

Figure 5.3: Box plot for VR times

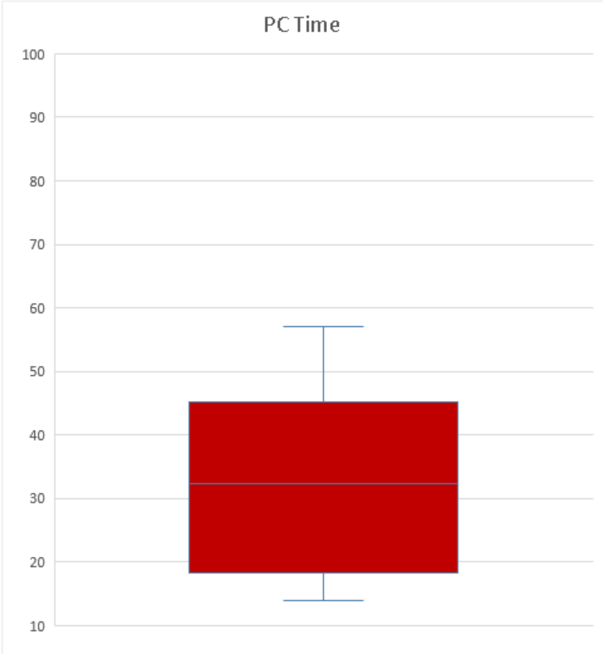

Figure 5.4: Box plot for PC times

device with controllers. The participants interacted with the visualization tool moving their heads and using the VR controllers.

Each phase started with the explanation of the task at hand, a basic tutorial on how to use the tools and a short adaptation time to the application. At the end of each phase the participants were asked to answer a questionnaire where they rated their perceived difficulty of the task and general happiness performing it. The time and correctness of the completed task were also measured.

5.2.5 Results

We quantitatively analyzed the results of the controlled experiment based on the statistical analysis of the collected data. We started by generating a box diagram for both the measured Phase A times (Fig.5.4) and for the measured Phase B times (Fig.5.3) to exclude outliers. From this process entry 6 of [C.2] was no longer considered in further analysis.

Table 5.6 shows the results of the statistical tests that we carry out to analyze the measured variable of user performance (completion time and correctness) and user experience (difficulty, cybersickness and enjoyment).

Task	Completion Time (s)		Difficulty		VR Enjoy		Correctness	Cybersickness
	Mean	SD	Mean	SD	Mean	SD	% Success	% of participants who felt it
Phase A	22,3	9,2	2,1	1,7	N/A	N/A	100	N/A
Phase B	24,7	6,3	2,6	1,1	4,7	0,5	100	14,3

Table 5.6: Summary of the results in terms of completion time, correctness and perceived difficulty

5.2.6 Discussion

This study provides a practical feedback on the prototype developed. In terms of completion time it can be concluded that the mean time completion of the task when performed in the PC is slightly lower than when performed on VR. Although we could mention that the difference is minor (2,4 seconds), that, given the standard deviation (SD) calculated, we could conclude that some participants were indeed faster in VR, or even that the participants were way more familiar with the PC environment when compared to VR, given the concrete data, we have to reject hypothesis H1. Similar arguments could be used to justify the rejection of hypothesis H3. The data supports the acceptance of hypothesis H2, but the possible low difficulty of the task performed (correctness of 100% in both) does not provide a very useful statistic.

In terms of cybersickness, only 14.3% of respondents reported feeling some of the symptoms (mostly due to the joystick movement). The perception of enjoyment was the field with better consensus among the group with a mean of 4.7 out of 5 (where 5 means extremely positive) and a standard deviation of 0.7.

As an overall summary of this experiment, even considering the rejection of two of the hypotheses initially raised, the results are not discouraging, as we believe the novelty of VR (and inexperience of the group with it) might have affected the performance of the group, suggesting the existence of a margin for progress as users adapt to the technology.

5.3 Interactions Evaluation

Since the prototype couldn't be tested by entities on the field of biophysics and since those are the main target of the proposal evaluated in this dissertation we decided to create a theoretical evaluation of the prototype based on a video[1] showcasing the interactions available, how to perform them and its effect on the scene. The target audience to which this questionnaire was purposed contained mostly biophysical researchers - clients of the partner company - and software developers - workers of the partner company - and were chosen due to their proximity and knowledge with the needs of the field. The goal of this test is to make sure the set of interactions available is clear and answers the call of computational biophysics needs in a natural way. We also wanted to get some feedback if the implementation of alternative techniques would be welcome.

We agree that this evaluation is not ideal but taking into account the working conditions to which we were subject to, by the pandemic situation we were living at the time of writing, it was the best we could do.

5.3.1 Participants

This evaluation was performed at a later stage of the period dedicated to this dissertation (since it was dependent on the completion of the prototype and therefore the form created [Appendix B.1] contains a sample size that is smaller than expected (only 6 persons). All participants were

[1] https://youtu.be/LN0IB1WC_0k

Attributes	Sub-Groups	Frequency(N=6)	Percentage (%)
Gender	Male	6	100
Age Group	31-50	6	100
Education Level	Bachelor	1	16.7
	Graduate	4	66.7
	Doctoral	1	16.7
Occupation	Project Manager	1	16.7
	Software Engineer/Developer (for scientific related projects)	5	83.3
VR Experience	None	1	16.7
	Less than 10 experiences	4	66.7
	10-100 experiences	1	16.7
Experience of visualizing biophysical data	None	3	50
	Less than 10 experiences	1	16.7
	10-100 experiences	1	16.7
	Over 100 experiences	1	16.7

Table 5.7: Sociodemographic factors and participants expertise

male, between 31-50, with college education and developers, project managers or clients of the partner company (MetaCell). 66.7% of the respondents had less than 10 VR experiences, 16.7% had between 10 and 100 and 16.7% as well had no experience at all. 50% had no experience visualizing biophysical data and the other half was evenly distributed between less than 10 experiences (16.7%), between 10 and 100 experiences (16.7%) and over 100 experiences (16.7%). This information is available in the form of table in 5.7.

5.3.2 Results

We tried to interpret the results of the form by averaging the score each entity gave to the linear choice questions (1-5) and calculate the percentage of yes's on yes/no questions. Table 5.7 shows a summary of the results.

5.3.3 Discussion

If in one hand 5.2 provided feedback on the prototype from a practical point of view, on the other hand this study provides a theoretical over it. Overall, it seems that the participants found the interactions clear and natural, as every question regarding those matters got an average result not lower than 50%. It become clear to us that users tend to prefer interactions based on gestures over button presses, as the 'bring closer' interaction (the only interaction performed with a typical button) rated last in terms of natural perception and is the one with lowest positive impression (3.8 out of 5, where 5 means extremely positive). Another interesting conclusion, which concerns the movement technique, is the tendency to prefer real movement techniques over the use of the controller's thumbstick to perform the navigation (83.3% believe it would be more productive to walk instead of use the joystick). Although we can see it's a debatable idea (as it contains

Question	Linear Scale (1-5)	Percentage (%)
Movement Impression	4,2	
Movement Clear		100
Movement Natural		100
Real Movement Productivity		83,3
Real Movement Ergonomics		66,7
Selection Impression	4	
Selection Clear		100
Selection Natural		83,3
Selection Errors	2,8	
Rotation Impression	4,2	
Rotation Clear		100
Rotation Natural		100
Scaling Impression	4	
Scaling Clear		100
Scaling Natural		83,3
Bring Closer Impression	3,8	
Bring Closer Clear		83,3
Bring Closer Natural		50

Table 5.8: Summary of the feedback for the prototype interactions

the second lowest percentage in the form replies), the concerns in terms of ergonomics for the implementation of real movement techniques seem to be lower than the ones we expected. A final outcome that we can retrieve from the results is the expectation of having less selection errors (2.8 out of 5, where 5 was more errors) with VR than with a PC; difficulties in selection is a concerned typically pointed out against 3D visualization that we wanted to get rid of with the VR implementation.

This feedback can have an extreme importance for future evolutions of the prototype and for its implementation in real computational biophysics tools.

5.3.4 Summary

In this section we described and discussed the studies performed to evaluate the the acceptance of VR into computational biophysics the advantages of using the VR prototype over the standard 3D version and the naturalness and clarity of the interactions.

We were able to retrieve important feedback, namely regarding the need to invest in showing to the entities of the field of computational biophysics the usefulness of VR; the importance of the factor of 'habituation' to a given technology in terms of measuring performance and the theoretical preference for gestures and more natural techniques for VR interactions.

The following chapter will conclude the dissertation by summarizing our learnings during this journey and pinpoint future directions.

Chapter 6

Conclusions and Future Work

Throughout the elaboration of this dissertation we tried to answer the question:

> "Would the introduction of VR in computational biophysics be a valuable and desired feature for and by the scientific researchers of today?"

This question led us to problems that went from the most basic VR implementations (navigation and object manipulation) to complex computational neuroscience modulations. The two areas covered by these problems, VR and computational neuroscience (a field of computational biophysics) have been the target of intense scientific activity, due to their individual potential. The junction of these two fields, driven by the evolution in the area of hardware that resulted in a rampant widespread use of VR headsets, may originate new revolutionary scientific discoveries. This idea is supported by the benefits associated with VR in terms of data visualization, in particular of 3D structures (as is the case with biophysical data). Within these, some can be highlighted:

- Easier navigation through the scene.
- More natural interactions.
- Improved spatial awareness.

The ultimate goal, in terms of applicability, would be the creation of a computational biophysics application that would have:

- The ability to easily adjust simulations through VR.
- The option to do multi-user collaborations.
- A natural mechanism for multi-scale navigation.

In the relatively short duration of the dissertation, it was possible to build a prototype that implements several methods of the most recent state of the art and perform some in-depth analysis on it. The evaluations carried out on the work performed, which were in part limited by the pandemic situation lived during the developing of this thesis, were not ideal but revealed some encouraging results that suggest the introduction of VR into computational biophysics would be valuable and valued for and by the entities of the field.

In the future, we foresee the addition of some new features to the work already accomplished, for example:

- Implement a real navigation method.
- Add collaboration between multiple users.
- Add multi-object selection.
- Add ability to perform real simulations.
- Add collisions between menu interactable objects (like a pipette) and the scene.
- Update current interactions to be gesture based.
- Make the application headset independent.
- Use hands tracking instead of VR Controllers.

We predict an auspicious future for projects develop in computational biophysics with VR and we truly believe that this junction will be a stepping stone into the next generation of computational biophysics.

Appendix A

TAM Form

A.1 Survey

7/2/2020 Advancing Computational Biophysics with Virtual Reality

Advancing Computational Biophysics with Virtual Reality

Form to evaluate the potential acceptance of using virtual reality to visualize and interact with computational scientific data, following the technology acceptance model (TAM) paradigm

*Obrigatório

Gender *

- ◯ Male
- ◯ Female

Age group *

- ◯ Under 18
- ◯ 18-30
- ◯ 31-50
- ◯ Over 50

Education level *

- ◯ Bachelor
- ◯ Graduate
- ◯ Doctoral
- ◯ Outra:

https://docs.google.com/forms/d/e/1FAIpQLSeu-atrTYd5mTbvtLPSjp93QucANvosJKCEcN-7FaT8cRI2hQ/viewform 1/6

Occupation *

- ◯ Bioinformatic Researcher
- ◯ Biophysical Researcher
- ◯ Software Engineer/Developer (for scientific related projects)
- ◯ Outra:

VR experience *

- ◯ None
- ◯ Less than 10 experiments
- ◯ 10 - 100 experiments
- ◯ Over 100 experiments

Experience of visualizing biophysical data *

- ◯ None
- ◯ Less than 10 experiments
- ◯ 10 - 100 experiments
- ◯ Over 100 experiments

Perceived Usefulness

The degree to which a person believes that using a particular system would enhance his/her job performance

[PU1] Using VR would improve biophysical scientific research productivity *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[PU2] The use of VR would make biophysical scientific research easier *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[PU3] Using VR would improve biophysical scientific research quality *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[PU4] Overall I would find the visualization of biophysical scientific data through VR useful for biophysical scientific research *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

Perceived Ease of Use
The degree to which a person believes that using a particular system would be free of effort

[E1] I would find computational biophysical scientific data visualized through VR easy to interact with

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[E2] I would find the information acquired through VR easy to understand *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[E3] I would find it easy to learn how to use VR *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[E4] I would find it easy to become skillful at using VR *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

Attitude Towards Using

The degree of evaluative affect that an individual associates with using a system in his or her job

[A1] My impression of using VR is *

	1	2	3	4	5	
Extremely negative	○	○	○	○	○	Extremely positive

Behavioral Intention

The degree to which a person has formulated conscious plans to perform or not perform some specified future behavior

[BI1] I would consider use VR to visualize and interact with computational biophysical scientific data *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[BI2] I intend to use VR to visualize and interact with computational biophysical scientific data within the foreseeable future *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[BI3] I will use VR to visualize and interact with biophysical computational scientific data within the foreseeable future *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

[BI4] I would recommend the use of VR to others for biophysical computational scientific data visualization *

	1	2	3	4	5	
Extremely disagree	○	○	○	○	○	Extremely agree

Submeter

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. Denunciar abuso - Termos de Utilização - Política de privacidade

Google Formulários

A.2 Survey Results

$PU1$	$PU2$	$PU3$	$PU4$	$E1$	$E2$	$E3$	$E4$	$A1$	$BI1$	$BI2$	$BI3$	$BI4$
4.0	5.0	4.0	5.0	5.0	4.0	5.0	5.0	3.0	4.0	3.0	3.0	5.0
4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	5.0	4.0	4.0	4.0	4.0
3.0	3.0	2.0	5.0	4.0	3.0	4.0	4.0	4.0	4.0	4.0	4.0	3.0
5.0	5.0	5.0	5.0	4.0	5.0	5.0	3.0	5.0	5.0	1.0	1.0	3.0
3.0	3.0	3.0	3.0	2.0	4.0	5.0	5.0	4.0	4.0	2.0	2.0	4.0
3.0	3.0	4.0	4.0	4.0	4.0	5.0	5.0	4.0	5.0	3.0	2.0	3.0
5.0	5.0	5.0	5.0	5.0	1.0	1.0	1.0	1.0	3.0	1.0	1.0	5.0
4.0	3.0	4.0	3.0	3.0	4.0	4.0	4.0	4.0	5.0	3.0	2.0	3.0
5.0	5.0	5.0	5.0	3.0	5.0	4.0	5.0	5.0	5.0	3.0	3.0	3.0
4.0	4.0	5.0	5.0	4.0	4.0	4.0	4.0	5.0	5.0	5.0	5.0	5.0
3.0	3.0	4.0	4.0	5.0	5.0	3.0	3.0	2.0	4.0	4.0	4.0	4.0
4.0	4.0	4.0	4.0	5.0	5.0	5.0	4.0	4.0	4.0	4.0	1.0	4.0
4.0	3.0	4.0	4.0	2.0	4.0	3.0	2.0	4.0	5.0	4.0	4.0	4.0
3.0	4.0	2.0	3.0	3.0	3.0	3.0	3.0	3.0	4.0	2.0	2.0	2.0
3.0	3.0	3.0	4.0	5.0	5.0	5.0	5.0	4.0	4.0	3.0	3.0	3.0

Table A.1: TAM questionnaire results.

Appendix B

Interactions Form

B.1 Survey

7/2/2020 Advancing Computational Biophysics with Virtual Reality

Advancing Computational Biophysics with Virtual Reality

Form to evaluate the clarity and naturality of the interactions implemented in the prototype. Please make sure to watch the introductory video here (https://youtu.be/LN0IB1WC_0k) before answering.

*Obrigatório

Gender *

- Male
- Female

Age group

- Under 18
- 18-30
- 31-50
- Over 50

https://docs.google.com/forms/d/e/1FAIpQLScSAmcZYdrLgYgWBbLKpTPczVuLD7DWLH0n9j3ybQOQQEMOEw/viewform 1/8

Education level

- ◯ Associate
- ◯ Bachelor
- ◯ Graduate
- ◯ Doctoral
- ◯ Outra:

Occupation

- ◯ Research Scientist
- ◯ Bioinformatic Scientist
- ◯ Neurologist
- ◯ Software Engineer - Data Visualization Engineer
- ◯ Software Engineer/Developer (for scientific related projects)
- ◯ Outra:

VR experience

- ◯ None
- ◯ Less than 10 experiences
- ◯ 10 - 100 experiences
- ◯ Over 100 experiences

Experience of visualizing biophysical data

- ◯ None
- ◯ Less than 10 experiences
- ◯ 10 - 100 experiences
- ◯ Over 100 experiences

Movement

Headset real rotation + thumbstick movement

My impression of the current movement implementation was *

	1	2	3	4	5	
Extremely negative	◯	◯	◯	◯	◯	Extremely positive

Is it clear how to move around the scene? *

- ◯ Yes
- ◯ No

Do you believe the current movement implementation feels natural? *

- ◯ Yes
- ◯ No

Would you say that using strategies that rely on real world movement (no use of joystick) would be preferable in terms of productivity?

- ◯ Yes
- ◯ No

Do you think that using strategies that rely on real world movement would be ergonomically feasible for companies?

- ◯ Yes
- ◯ No

Selection

Raycaster + trigger button

My impression of the selection procedure was *

	1	2	3	4	5	
Extremely negative	◯	◯	◯	◯	◯	Extremely positive

Is it clear how to perform the selection? *

- ◯ Yes
- ◯ No

Do you believe the selection procedure feels natural? *

- Yes
- No

Would you say that the selection implementation is prone to less/same/more errors than the typical mouse selection in standard 3D scenes?

	1	2	3	4	5	
Less Errors	◯	◯	◯	◯	◯	More Errors

Rotation

Grip button + Controller rotation

My impression of the rotation action was *

	1	2	3	4	5	
Extremely negative	◯	◯	◯	◯	◯	Extremely positive

Is it clear how to rotate objects? *

- Yes
- No

Do you believe the rotation action feels natural? *

- ◯ Yes
- ◯ No

Scaling
Grip button + Grip button + Controllers gestures

My impression of the scaling gesture was *

	1	2	3	4	5	
Extremely negative	◯	◯	◯	◯	◯	Extremely positive

Is it clear how to scale objects? *

- ◯ Yes
- ◯ No

Do you believe the scaling gesture feels natural? *

- ◯ Yes
- ◯ No

Bring it closer
Press A to bring the model / selected objects closer

My impression of the 'bring it closer' action was *

	1	2	3	4	5	
Extremely negative	○	○	○	○	○	Extremely positive

Is it clear how to bring an object closer? *

- ○ Yes
- ○ No

Do you believe the 'bring it closer' action feels natural? *

- ○ Yes
- ○ No

Conclusions

Please type below any feedback you would like to give on this prototype (new interactions, changes for the current interactions, etc...)

Feedback

A sua resposta

Submeter

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. Denunciar abuso - Termos de Utilização - Política de privacidade

B.2 Survey Results

$PU1$	$PU2$	$PU3$	$PU4$	$E1$	$E2$	$E3$	$E4$	$A1$	$BI1$	$BI2$	$BI3$	$BI4$
4.0	5.0	4.0	5.0	5.0	4.0	5.0	5.0	3.0	4.0	3.0	3.0	5.0
4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	5.0	4.0	4.0	4.0	4.0
3.0	3.0	2.0	5.0	4.0	3.0	4.0	4.0	4.0	4.0	4.0	4.0	3.0
5.0	5.0	5.0	5.0	4.0	5.0	5.0	3.0	5.0	5.0	1.0	1.0	3.0
3.0	3.0	3.0	3.0	2.0	4.0	5.0	5.0	4.0	4.0	2.0	2.0	4.0
3.0	3.0	4.0	4.0	4.0	4.0	5.0	5.0	4.0	5.0	3.0	2.0	3.0
5.0	5.0	5.0	5.0	5.0	1.0	1.0	1.0	1.0	3.0	1.0	1.0	5.0
4.0	3.0	4.0	3.0	3.0	4.0	4.0	4.0	4.0	5.0	3.0	2.0	3.0
5.0	5.0	5.0	5.0	3.0	5.0	4.0	5.0	5.0	5.0	3.0	3.0	3.0
4.0	4.0	5.0	5.0	4.0	4.0	4.0	4.0	5.0	5.0	5.0	5.0	5.0
3.0	3.0	4.0	4.0	5.0	5.0	3.0	3.0	2.0	4.0	4.0	4.0	4.0
4.0	4.0	4.0	4.0	5.0	5.0	5.0	4.0	4.0	4.0	4.0	1.0	4.0
4.0	3.0	4.0	4.0	2.0	4.0	3.0	2.0	4.0	5.0	4.0	4.0	4.0
3.0	4.0	2.0	3.0	3.0	3.0	3.0	3.0	3.0	4.0	2.0	2.0	2.0
3.0	3.0	3.0	4.0	5.0	5.0	5.0	5.0	4.0	4.0	3.0	3.0	3.0

Table B.1: Interactions questionnaire results.

Appendix C

Tests Form

C.1 Survey

Advancing Computational Biophysics with Virtual Reality

Form to compare the Virtual Reality implementation with the PC implementation in terms of navigation and interactions

*Obrigatório

Gender *

- ◯ Male
- ◯ Female

Age group *

- ◯ Under 18
- ◯ 18-30
- ◯ 31-50
- ◯ Over 50

Education level *

- ◯ Bachelor
- ◯ Graduate
- ◯ Doctoral
- ◯ Outra:

Occupation *

- ◯ Research Scientist
- ◯ Bioinformatic Scientist
- ◯ Neurologist
- ◯ Software Engineer - Data Visualization Engineer
- ◯ Software Engineer/Developer (for scientific related projects)
- ◯ Outra:

VR experience *

- ◯ None
- ◯ Less than 10 experiences
- ◯ 10 - 100 experiences
- ◯ Over 100 experiences

Experience of visualizing biophysical data *

- ◯ None
- ◯ Less than 10 experiences
- ◯ 10 - 100 experiences
- ◯ Over 100 experiences

Finding the correct object in the PC version was *

	1	2	3	4	5	
Extremely easy	◯	◯	◯	◯	◯	Extremely hard

Finding the correct object in the VR version was *

	1	2	3	4	5	
Extremely easy	◯	◯	◯	◯	◯	Extremely hard

Did you feel any discomfort when using VR?

◯ Yes

◯ No

My impression of using VR was *

	1	2	3	4	5	
Extremely negative	◯	◯	◯	◯	◯	Extremely positive

Correct object selected VR?

◯ Yes

◯ No

Time taken (seconds) VR

A sua resposta

Correct object selected PC?

- ◯ Yes
- ◯ No

Time taken (seconds) PC

A sua resposta

Submeter

Nunca envie palavras-passe através dos Google Forms.

Este conteúdo não foi criado nem aprovado pela Google. Denunciar abuso - Termos de Utilização - Política de privacidade

Google Formulários

C.2 Survey Results

	PC Difficulty	VR Difficulty	CyberSickness	VR Enjoy	VR Correct	VR Time	PC Correct	PC Time
Participant 1	1	2	Yes	5	Yes	29	Yes	20
Participant 2	4	4	No	5	Yes	30	Yes	22
Participant 3	5	3	No	5	Yes	32	Yes	39
Participant 4	1	2	No	5	Yes	18	Yes	15
Participant 5	1	4	No	4	Yes	27	Yes	16
Participant 6	1	5	Yes	4	No	600	Yes	111
Participant 7	1	2	No	4	Yes	21	Yes	14
Participant 8	2	1	No	5	Yes	16	Yes	30

Table C.1: Tests questionnaire results.

References

[1] Christoph Anthes, Rubén García Hernandez, Markus Wiedemann, and Dieter Kranzlmüller. State of the art of virtual reality technologies. 03 2016.

[2] Mohamad ASK and Mohsin Reza. Computational models in neuroscience: How real are they? a critical review of status and suggestions. *Austin Neurology amp; Neurosciences*, 2016.

[3] Richard P Bagozzi and Youjae Yi. On the evaluation of structural equation models. *Journal of the academy of marketing science*, 16(1):74–94, 1988.

[4] Alex Banks and Eve Porcello. *Learning React: functional web development with React and Redux*. " O'Reilly Media, Inc.", 2017.

[5] David Beeman. A modeling study of cortical waves in primary auditory cortex. *BMC Neuroscience*, 14(S1):P23, 2013.

[6] José M. Bernardo and Raúl Rueda. Bayesian hypothesis testing: a reference approach. *International Statistical Review*, 70(3):351–372, 2002.

[7] James M Bower and David Beeman. Genesis (simulation environment). *Scholarpedia*, 2(3):1383, 2007.

[8] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 205–216, 2016.

[9] Nell Waliczek Brandon Jones. Webxr device api. https://www.w3.org/TR/webxr/. Accessed: 05/07/2020.

[10] Richard Brath. 3d infovis is here to stay. 11 2014.

[11] Frederick Brooks, Jr. What's real about virtual reality? *IEEE Computer Graphics and Applications*, 19:16–27, 12 1999.

[12] Don Brutzman and Leonard Daly. *X3D: extensible 3D graphics for Web authors*. Elsevier, 2010.

[13] Steve Bryson. Approaches to the successful design and implementation of vr applications. *Virtual reality applications*, pages 3–15, 1995.

[14] Steve Bryson. Virtual reality in scientific visualization. *Communications of the ACM*, 39(5):62–71, 1996.

[15] Ricardo Cabello et al. Three. js. https://github.com/mrdoob/three.js, 34, 2010. Accessed: 05/07/2020.

[16] Robert C Cannon, Marc-Oliver Gewaltig, Padraig Gleeson, Upinder S Bhalla, Hugo Cornelis, Michael L Hines, Fredrick W Howell, Eilif Muller, Joel R Stiles, Stefan Wils, et al. Interoperability of neuroscience modeling software: current status and future directions. *Neuroinformatics*, 5(2):127–138, 2007.

[17] Robert C Cannon, Padraig Gleeson, Sharon Crook, Gautham Ganapathy, Boris Marin, Eugenio Piasini, and R Angus Silver. Lems: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning neuroml 2. *Frontiers in neuroinformatics*, 8:79, 2014.

[18] Matteo Cantarelli, Boris Marin, Adrian Quintana, Matt Earnshaw, Robert Court, Padraig Gleeson, Salvador Dura-Bernal, R Angus Silver, and Giovanni Idili. Geppetto: a reusable modular open platform for exploring neuroscience data and models. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1758):20170380, 2018.

[19] C. Chen. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, 25(4):12–16, July 2005.

[20] Min Chen, Luciano Floridi, and Rita Borgo. What is visualization really for? *The Philosophy of Information Quality*, page 75–93, 2014.

[21] Patricia S Churchland and Terrence J Sejnowski. *The computational brain*. MIT press, 2016.

[22] Thiago Vilela Cruz. Virtual reality in the architecture, engineering and construction industry proposal of an interactive collaboration application. 2018.

[23] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[24] Alexandre Valle de Carvalho. Spatio-temporal information management and visualization. https://hdl.handle.net/10216/57985, 2009. FEUP. PhD Thesis.

[25] Thomas A. Defanti and Maxine D. Brown. Visualization in scientific computing. volume 33 of *Advances in Computers*, pages 247 – 307. Elsevier, 1991.

[26] Christian Dibbern, Manuela Uhr, Dennis Krupke, and Frank Steinicke. Can webvr further the adoption of virtual reality? *Mensch und Computer 2018-Usability Professionals*, 2018.

[27] El Bachir Diop, Shengchuan Zhao, and Tran Van Duy. An extension of the technology acceptance model for understanding travelers' adoption of variable message signs. *PLoS one*, 14(4):e0216007, 2019.

[28] Alessandra Donato, Konstantinos Kagias, Yun Zhang, and Massimo A. Hilliard. Neuronal sub-compartmentalization: a strategy to optimize neuronal function. *Biological Reviews*, 94(3):1023–1037, 2019.

[29] Mohammed Elbamby, Cristina Perfecto, Mehdi Bennis, and Klaus Doppler. Toward low-latency and ultra-reliable virtual reality. *IEEE Network*, 32, 01 2018.

[30] Claes Fornell and David F Larcker. Evaluating structural equation models with unobservable variables and measurement error. *Journal of marketing research*, 18(1):39–50, 1981.

[31] D. Freeman, Sarah Reeve, A. Robinson, Anke Ehlers, David Clark, Bernhard Spanlang, and Mel Slater. Virtual reality in the assessment, understanding, and treatment of mental health disorders. *Psychological Medicine*, 47:1–8, 03 2017.

[32] Tom C Freeman, Leon Goldovsky, Markus Brosch, Stijn Van Dongen, Pierre Mazière, Russell J Grocock, Shiri Freilich, Janet Thornton, and Anton J Enright. Construction, visualisation, and clustering of transcription networks from microarray expression data. *PLoS Comput Biol*, 3(10):e206, 2007.

[33] Fabrizio Galeazzi, Marco Callieri, Matteo Dellepiane, Michael Charno, Julian Richards, and Roberto Scopigno. Web-based visualization for 3d data in archaeology: The ads 3d viewer. *Journal of Archaeological Science: Reports*, 9:1 – 11, 2016.

[34] Sergio E Galindo, Pablo Toharia, Oscar D Robles, and Luis Pastor. Visimpl: multi-view visual analysis of brain simulation data. *Frontiers in Neuroinformatics*, 10:44, 2016.

[35] Elliot Gardner. Unreal engine: from gaming to ground-breaking cures, Nov 2018.

[36] Padraig Gleeson, Sharon Crook, Robert C Cannon, Michael L Hines, Guy O Billings, Matteo Farinella, Thomas M Morse, Andrew P Davison, Subhasis Ray, Upinder S Bhalla, et al. Neuroml: a language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Comput Biol*, 6(6):e1000815, 2010.

[37] Dan Goodman and Romain Brette. The brian simulator. *Frontiers in neuroscience*, 3:192–7, 09 2009.

[38] Joseph F Hair, William C Black, Barry J Babin, Rolph E Anderson, Ronald L Tatham, et al. *Multivariate data analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 1998.

[39] Kenneth P. Herndon, Andries van Dam, and Michael Gleicher. The challenges of 3d interaction: A chi '94 workshop. *SIGCHI Bull.*, 26(4):36–43, October 1994.

[40] Michael Hines. Neuron—a program for simulation of nerve equations. In *Neural systems: Analysis and modeling*, pages 127–136. Springer, 1993.

[41] A. L. HODGKIN and A. F. HUXLEY. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. (Lond.)*, 117(4):500–544, Aug 1952.

[42] John Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–8, 05 1982.

[43] D. H. HUBEL and T. N. WIESEL. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol. (Lond.)*, 160:106–154, Jan 1962.

[44] David Hunter Hubel and Torsten Nils Wiesel. Ferrier lecture-functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 198(1130):1–59, 1977.

[45] Geoffrey S Hubona, Gregory W Shirah, and David G Fout. 3d object recognition with motion. In *CHI'97 extended abstracts on Human factors in computing systems*, pages 345–346. 1997.

[46] Thomas Huk. Who benefits from learning with 3d models? the case of spatial ability. *Journal of computer assisted learning*, 22(6):392–404, 2006.

[47] Arnim Jenett, Gerald M Rubin, Teri-TB Ngo, David Shepherd, Christine Murphy, Heather Dionne, Barret D Pfeiffer, Amanda Cavallaro, Donald Hall, Jennifer Jeter, et al. A gal4-driver line resource for drosophila neurobiology. *Cell reports*, 2(4):991–1001, 2012.

[48] R.S. Kalawsky, J. O'Brien, and P.V. Coveney. Improving scientists' interaction with complex computational–visualization environments based on a distributed grid infrastructure. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 363(1833):1867–1884, Aug 2005.

[49] Rabimba Karanjai. *Optimizing Web Virtual Reality*. PhD thesis, 2018.

[50] David A Kenny. Measuring model fit. 2012. 17, 2018.

[51] G. Korakakis, Andreas Boudouvis, J.A. Palyvos, and Evangelia Pavlatou. The impact of 3d visualization types in instructional multimedia applications for teaching science. *Procedia - Social and Behavioral Sciences*, 31:145–149, 12 2012.

[52] Michal Koutek. Scientific visualization in virtual reality: interaction techniques and application development. 01 2003.

[53] Roman Kucherenko. Webvr api description and a-frame application implementation. 2019.

[54] Steven M. LaValle. *Virtual Reality*. Cambridge University Press, 2020.

[55] Jon Leech and Benj Lipchak. *OpenGL ES*. The Khronos Group Inc, 2019.

[56] Ed S Lein, Michael J Hawrylycz, Nancy Ao, Mikael Ayres, Amy Bensinger, Amy Bernard, Andrew F Boe, Mark S Boguski, Kevin S Brockway, Emi J Byrnes, et al. Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, 2007.

[57] William W Lytton. *From computer to brain: foundations of computational neuroscience*. Springer Science & Business Media, 2007.

[58] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, April 1986.

[59] Kerry Manis and Danny Choi. The virtual reality hardware acceptance model (vr-ham): Extending and individuating the technology acceptance model (tam) for virtual reality hardware. *Journal of Business Research*, 10 2018.

[60] David Marr, David Willshaw, and Bruce McNaughton. Simple memory: a theory for archicortex. In *From the Retina to the Neocortex*, pages 59–128. Springer, 1991.

[61] Kouichi Matsuda and Rodger Lea. Webgl programming guide: Interactive 3d graphics programming with webgl. 2013.

[62] Leonel Merino, Alexandre Bergel, and Oscar Nierstrasz. Overcoming issues of 3d software visualization through immersive augmented reality. 09 2018.

[63] Michele Migliore, Michele Ferrante, and Giorgio A Ascoli. Signal propagation in oblique dendrites of ca1 pyramidal cells. *Journal of neurophysiology*, 94(6):4145–4155, 2005.

[64] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, volume 2351, pages 282–292. International Society for Optics and Photonics, 1995.

[65] Nestor Milyaev, David Osumi-Sutherland, Simon Reeve, Nicholas Burton, Richard A. Baldock, and J. Douglas Armstrong. The Virtual Fly Brain browser and query interface. *Bioinformatics*, 28(3):411–415, 12 2011.

[66] Ali A. Minai. Computational neuroscience : A brief overview. 2007.

[67] Meghan C Mott, Joshua A Gordon, and Walter J Koroshetz. The nih brain initiative: Advancing neurotechnologies, integrating disciplines. *PLoS biology*, 16(11):e3000066, 2018.

[68] Samuel A Neymotin, Dylan S Daniels, Blake Caldwell, Robert A McDougal, Nicholas T Carnevale, Mainak Jas, Christopher I Moore, Michael L Hines, Matti Hämäläinen, and Stephanie R Jones. Human neocortical neurosolver (hnn), a new software tool for interpreting the cellular and network origin of human meg/eeg data. *Elife*, 9:e51214, 2020.

[69] Takanori Okoshi. *Three-dimensional imaging techniques*. Elsevier, 2012.

[70] G. Scott Owen. Hypervis - teaching scientific visualization using hypermedia. *ACM SIG GRAPH*, 1999.

[71] Sylvia Xueni Pan. Interacting with objects in vr. https://www.coursera.org/learn/3d-interaction-design-virtual-reality/home/week/3. University of London Goldsmiths. Coursera, Accessed: 05/07/2020.

[72] Tony Parisi. *Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages*. O'Reilly Media, Inc., 1st edition, 2014.

[73] Georgios A Pavlopoulos, Seán I O'Donoghue, Venkata P Satagopam, Theodoros G Soldatos, Evangelos Pafilis, and Reinhard Schneider. Arena3d: visualization of biological networks in 3d. *BMC systems biology*, 2(1):104, 2008.

[74] Hanchuan Peng, Zongcai Ruan, Fuhui Long, Julie H Simpson, and Eugene W Myers. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology*, 28(4):348–353, 2010.

[75] Alexander Peyser, Rajalekshmi Deepu, Jessica Mitchell, Shailesh Appukuttan, Till Schumann, Jochen Martin Eppler, David Kappel, Jan Hahne, Barna Zajzon, Itaru Kitayama, et al. Nest 2.14. 0. Technical report, Jülich Supercomputing Center, 2017.

[76] Krzysztof Pietroszek. Raycasting in virtual reality, 2019.

[77] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, 1996.

[78] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. *Redirected walking*. Citeseer, 2005.

[79] T. Rhyne, M. Tory, T. Munzner, M. Ward, C. Johnson, and D. H. Laidlaw. Information and scientific visualization: separate but equal or happy together at last. In *IEEE Visualization, 2003. VIS 2003.*, pages 611–614, 2003.

[80] Bernhard E Riecke, Bobby Bodenheimer, Timothy P McNamara, Betsy Williams, Peng Peng, and Daniel Feuereissen. Do we need to walk for effective virtual reality navigation? physical rotations alone may suffice. In *International Conference on Spatial Cognition*, pages 234–247. Springer, 2010.

[81] A Rodriguez, J Whitson, and R Granger. Derivation and analysis of basic computational operations of thalamocortical circuits. *Journal of cognitive neuroscience*, 16(5):856–877, 2004.

[82] Ilan Samish, Philip E. Bourne, and Rafael J. Najmanovich. Achievements and challenges in structural bioinformatics and computational biophysics. *Bioinformatics*, 31(1):146–150, 12 2014.

[83] Erik De Schutter. Computational neuroscience organization. `https://www.cnsorg.org/computational-neuroscience`. Accessed: 05/07/2020.

[84] Terrence J Sejnowski, Christof Koch, and Patricia Smith Churchland. Computational neuroscience. *Science*, 241(4871):1299–1306, 1988.

[85] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[86] Mel Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364:3549–57, 12 2009.

[87] Rebecca R. Springmeyer, Meera M. Blattner, and Nelson L. Max. A characterization of the scientific data analysis process. In *Proceedings of the 3rd Conference on Visualization '92*, VIS '92, page 235–242, Washington, DC, USA, 1992. IEEE Computer Society Press.

[88] Detlev Stalling, Malte Westerhoff, Hans-Christian Hege, et al. Amira: A highly interactive system for visual data analysis. *The visualization handbook*, 38:749–67, 2005.

[89] David Sterratt, Bruce Graham, Andrew Gillies, and David Willshaw. *Principles of computational modelling in neuroscience*. Cambridge University Press, 2011.

[90] Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42, 07 2000.

[91] E.A. Styles. The psychology of attention: Second edition. *The Psychology of Attention: Second Edition*, pages 1–336, 01 2006.

[92] Nikolai Suslov. Livecoding. space: Towards p2p collaborative live programming environment for webxr. In *Proceedings of the Fourth International Conference on Live Coding (ICLC 2019), Medialab Prado, Madrid, Spain, http://iclc. livecodenetwork. org/2019/papers/paper133. pdf*, 2019.

[93] Youssef Tanagho, Gerald Andriole, Alethea Paradis, Kerry Madison, Gurdarshan Sandhu, Esteban Varela, and Brian Benway. 2d versus 3d visualization: Impact on laparoscopic proficiency using the fundamentals of laparoscopic surgery skill set. *Journal of laparoendoscopic advanced surgical techniques. Part A*, 22, 10 2012.

[94] Wenjin Tao, Ze-Hao Lai, Ming C Leu, and Z Yin. Manufacturing assembly simulations in virtual and augmented reality. *Augmented, Virtual, and Mixed Reality Applications in Advanced Manufacturing*, 2018.

[95] Jeffery L Teeters, Keith Godfrey, Rob Young, Chinh Dang, Claudia Friedsam, Barry Wark, Hiroki Asari, Simon Peron, Nuo Li, Adrien Peyrache, et al. Neurodata without borders: creating a common data format for neurophysiology. *Neuron*, 88(4):629–634, 2015.

[96] Alfredo Teyseyre and Marcelo Campo. An overview of 3d software visualization. *IEEE transactions on visualization and computer graphics*, 15:87–105, 01 2009.

[97] R. A. Tikidji-Hamburyan, V. Narayana, Z. Bozkus, and T. A. El-Ghazawi. Software for Brain Network Simulations: A Comparative Study. *Front Neuroinform*, 11:46, 2017.

[98] Konrad Tollmar, David Demirdjian, and Trevor Darrell. Gesture + play: full-body interaction for virtual environments. pages 620–621, 01 2003.

[99] Thomas Trappenberg. *Fundamentals of computational neuroscience*. OUP Oxford, 2009.

[100] Martin Usoh, Kevin Arthur, Mary C Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P Brooks Jr. Walking> walking-in-place> flying, in virtual environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 359–364, 1999.

[101] Andries Van Dam, Andrew S Forsberg, David H Laidlaw, Joseph J LaViola, and Rosemary M Simpson. Immersive vr for scientific visualization: A progress report. *IEEE Computer Graphics and Applications*, 20(6):26–52, 2000.

[102] Andries Van Dam, David H Laidlaw, and Rosemary Michelle Simpson. Experiments in immersive virtual reality for scientific visualization. *Computers & Graphics*, 26(4):535–555, 2002.

[103] Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graph.*, 15(2):121–140, April 1996.

[104] Hsin-Kai Wu and Priti Shah. Exploring visuospatial thinking in chemistry learning. *Science Education*, 88:465–492, 05 2004.

[105] Quan Ye, Wenshan Hu, Hong Zhou, Zhongcheng Lei, and Shiqi Guan. Vr interactive feature of html5-based webvr control laboratory by using head-mounted display. *International Journal of Online Engineering (iJOE)*, 14:20, 03 2018.

[106] Difeng Yu, Hai-Ning Liang, Feiyu Lu, Vijayakumar Nanjappan, Konstantinos Papangelis, and Wei Wang. Target selection in head-mounted display virtual reality environments. *J. UCS*, 24(9):1217–1243, 2018.

[107] Shuguang Yuan, H. Chan, and Zhenquan Hu. Implementing webgl and html5 in macromolecular visualization and modern computer-aided drug design. *Trends in Biotechnolgoy*, 35, 04 2017.

[108] Michael Zyda and Tom DeFanti. Praise for understanding virtual reality: Interface, application, and design. In William R. Sherman and Alan B. Craig, editors, *Understanding Virtual Reality*, The Morgan Kaufmann Series in Computer Graphics, page i. Morgan Kaufmann, San Francisco, 2003.