Bachelor Degree Project

Using Leap Motion for the Interactive Analysis of Multivariate Networks

Author: Andreas LIF
Author: Marcello VENDRUSCOLO
Supervisor: Prof. Dr. Andreas KERREN
Examiner: Tobias OHLSSON
Semester: VT2020
Subject: Computer Science
Course code: 2DV50E

Abstract

This work is an interdisciplinary study involving mainly the fields of information visualisation and human-computer interaction. The advancement of technology has expanded the ways in which humans interact with machines, which has benefited both the industry as well as several fields within science. However, scientists and practitioners in the information visualisation domain remain working, mostly, with classical setups constituted of keyboard and standard computer mouse devices. This project investigates how a shift in the human-computer interaction aspect of visualisation software systems can affect the accomplishment of tasks and the overall user experience when analysing two-dimensionally displayed multivariate networks. Such investigation is relevant as complex network structures have seen an increase in use as essential tools to solve challenges that directly affect individuals and societies, such as in medicine or social sciences. The improvement of visualisation software's usability can result in more of such challenges answered in a shorter time or with more precision. To answer this question, a web application that enables users to analyse multivariate networks through interfaces based both on hand gesture recognition and mouse device was developed. Also, a number of gesture designs were developed for several tasks to be performed when visually analysing networks. Then, an expert in the field of human-computer interaction was invited to review the proposed hand gestures and report his overall user experience of using them. The results show that the expert had, overall, similar user experience for both hand gestures and mouse device. Moreover, the interpretation of the results indicates that the accuracy offered by gestures has to be carefully taken into account when designing gestures for selection tasks, particularly when the selection targets are small objects. Finally, our analysis points out that the manner in which the software's graphical user interface is presented also affects the usability of gestures, and that both factors have to be designed accordingly.

Keywords: Multivariate Networks, Information Visualisation, Hand Gesture Designs, Human-Computer Interaction, Natural User Interface, Leap Motion

Contents

List of Figures

List of Tables

1 Introduction

If you were contacted by a municipality to design a route which enables its citizens and visitors to traverse each of the bridges in the city exactly once but still finish the journey at the same location where it started, how would you approach the problem? This very challenge was presented to Leonhard Euler in the early 18th century. To solve such puzzle, he developed a mathematical representation of the city as a graph structure, creating the grounds of what is today known as graph theory [13]. Since then, mathematicians have explored and further developed the concepts within this field, generating knowledge and creating tools that have empowered the scientific community to solve problems of greater complexity. For example, graphs have enabled researchers to elucidate challenges in electrical engineering (e.g., communication networks), organic chemistry, biology and medicine (e.g., cellular networks and drug targets identification), and also sociology (e.g., social networks) [14, 15]. However, not only science has benefited; the software industry has also profited from graph concepts for the development of applications that are present in most people's everyday life, including Facebook, Instagram, Google Maps and Google search for web pages [16].

Despite the remarkable applicability of graphs in multidisciplinary areas of science and business projects, the use of such structures in the field of information visualisation requires constant development of supplementary technologies and their integration with existing visualisation software systems. The complexity of information stored in graph data structures can rapidly escalate, creating difficult challenges for researchers in the field of information visualisation [17, 18]. For example, if during the outbreak of novel coronavirus (COVID-19) you had to visually display the cities with highest contamination potential by analysing the worldwide air transportation network data, how would you illustrate such large data set? Would you only display part of the information, or perhaps distribute them in separate layers? Figure 1.1 gives an idea of the complexity and difficulties of such a task. Despite the diverse existing techniques in the field of information visualisation, including the ones just mentioned, such challenges are still not entirely solved [19]. However, it is important to remember that information visualisation reaches beyond just the visual composition of data; it involves the tasks that are executed by researchers and analysts investigating graph structures, and also the interaction styles employed [20].

Since the creation of the first computer devices, people have gone through different interactive experiences with machines as exploratory research in the field of human-computer interaction unfolded [21]. From command-line interfaces and pointing devices over direct manipulation of graphical objects to gesture and speech recognition interfaces and virtual reality, each interaction style offers unique benefits and downsides that make them more or less suitable for different tasks. In the field of information visualisation, traditional computer mouse devices have long been utilised by users to interact with software systems and execute tasks [22]. However, taking into consideration all the other existing manners of interaction, is mouse-guided interaction still the most optimal one to be used? Is it possible for the so crucial tasks of analysing and drawing conclusions from graphs to benefit from a change in the current most employed form of interaction? Would it somewhat alleviate the information visualisation challenges?

This research work is an interdisciplinary study which involves the previously mentioned fields of information visualisation (InfoVis), human-computer interaction (HCI), and graph drawing. It explores how a shift in the human-computer interaction facet of data visualisation can affect the overall user experience in the accomplishment of tasks

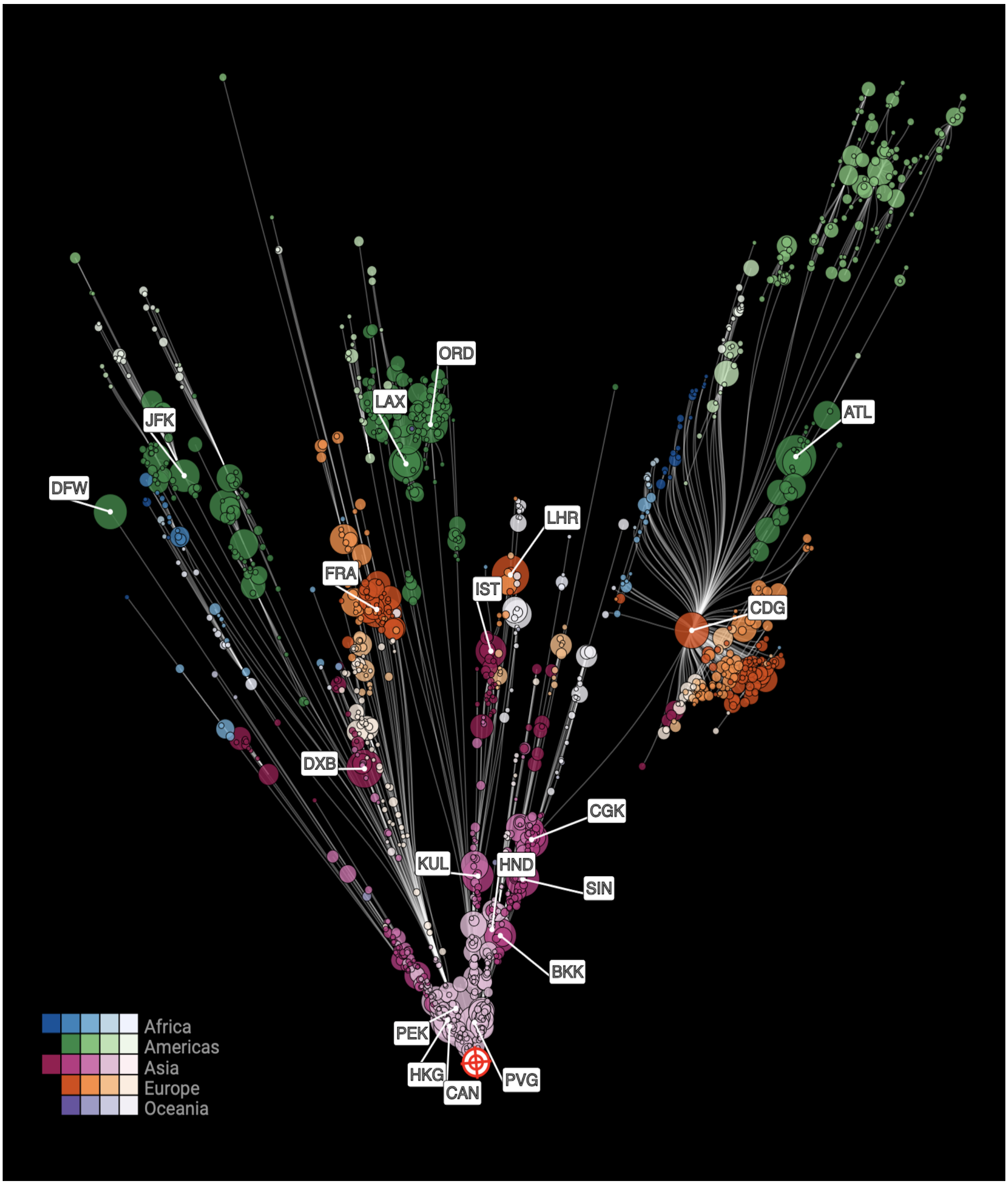


Figure 1.1: This figure shows the most likely spreading routes of the novel coronavirus (COVID-19) worldwide from Wuhan in China based on air transportation network [1]. It also provides information about the expected arrival time and effective distance. Due to the visualisation technique and properties of interest chosen, it can be difficult for one to distinguish information in dense areas of the drawing. This image depicts some of the information visualisation current challenges. This graph was produced through a collaborative effort between the Humboldt University of Berlin and the Robert Koch Institute in Berlin [2].

executed during the analysis of multivariate networks. In this context, it compares the effectiveness of the WIMP (Windows, Icons, Menus, and a Pointer) interface against the proposed interface, which is based on the recognition of hand gestures. The analysis of empirical data clarifies whether an improvement is observed in the overall usability and also indicates what tasks can benefit the most from such a change.

1.1 Background

This section contains explanations for the essential theories, findings, definitions and terms regarding graphs and multivariate networks, information visualisation, and human-computer interaction which are required to understand this project documentation. In addition, it contains an introduction to Leap Motion Controller, the technology which made possible the practical implementation of this project.

1.1.1 Graphs and Multivariate Networks

Graphs, also known as networks, are data structures composed of nodes and edges. Primordially, as employed by Euler to solve The Seven Bridges of Königsberg problem, the single purpose of nodes and edges was to represent entities and to indicate relationships between such entities, respectively. Therefore, considering a set of nodes, also known as vertices, V and a set of edges E, with $E \subseteq \{(u, v)|u, v \in V, u \neq v\}$, a simple graph G is mathematically defined as $G = (V, E)$ [23].

However, as science advanced and society needs progressed, data structures that could accommodate more complexity became necessary to support our ever-evolving systems. In contrast to a simple graph, a multivariate network (MVN) is an abstract network where nodes and edges, besides just illustrating entities and connections, also contain attributes concerning them [20]. The mathematical model of multivariate networks extends the definition of simple graphs by adding a collection of N attributes to the set of vertices, $a_n \in A_{vertices}, n \leq N$, and K attributes to the set of edges, $a_k \in A_{edges}, k \leq K$, as it can be observed in Figure 1.2 [3]. In such models, each vertice and edge has associated values for each attribute of its set.

The use of multivariate networks for representing complex domains, such as biological and environmental sciences, has grown more frequent mainly because such structures are capable of storing the increasing amounts of heterogeneous data generated in these domains. It is known that researchers and data analysts often repeat particular tasks with higher frequency when analysing multivariate networks and, although these sets of specific tasks normally differ from each other according to the domain under study, several research studies identify and classify the most common tasks found across analyses of graph data. Although the obtained results vary, Amar *et al.* identified a set of ten primitive tasks that are independent of visualisation systems and common in the analytic activity of data representing different domains [24]. In another study, Lee *et al.* extracted common tasks from various study cases of network visualisation techniques and the conclusion was that complicated tasks could all be accomplished by different combinations of the low-level tasks previously identified by Amar *et al.* together with three other primitive tasks introduced during the study [25]. The following list names and shortly describes each of these tasks for a given set of entities in a network:

- **Retrieve value:** get the value of attributes;
- **Filter:** narrow the set of entities according to constraints on attributes values;

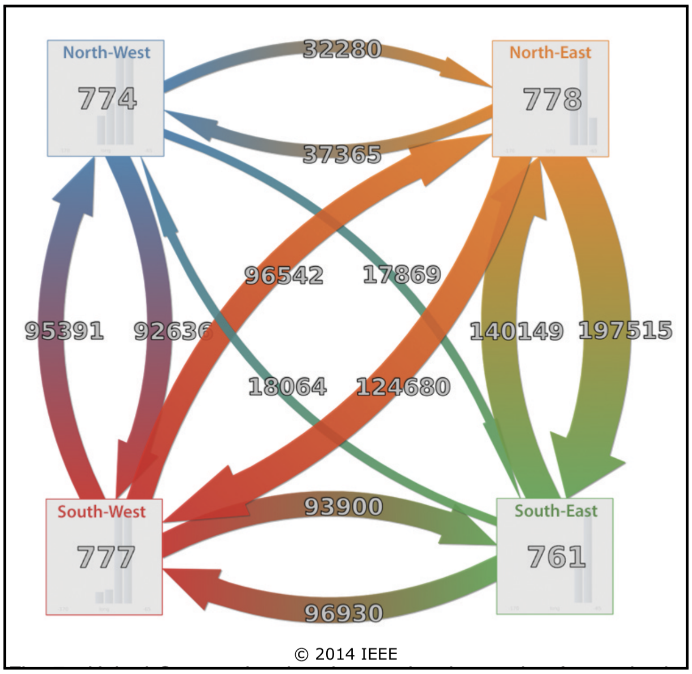


Figure 1.2: In the form of a multivariate network, this figure shows an overview of the migration in the United States of America. Notice that vertices and edges have attributes associated with them representing, respectively, the name and number of counties of each region and the inbound and outbound migration numbers. This figure is a cropped image retrieved from the research work presented by S. van Den Elzen and J. J. van Wijk [3].

- **Compute derived value:** numerically represent the set (e.g., count, average);
- **Find extremum:** get entities that hold maximum/minimum value of an attribute;
- **Sort:** arrange entities according to some criteria;
- **Determine range:** discover the extent of values for an attribute;
- **Characterise distribution:** define the distribution of values for an attribute;
- **Find annomalies:** recognise anomalies concerning an expectation or relationship;
- **Cluster:** discover dense areas where entities share similar values for attributes;
- **Correlate:** discover valuable relations among the values of two chosen attributes;
- **Find adjacent nodes:** discover vertices directly connected to a chosen node;
- **Scan:** quickly review a collection of entities; and
- **Set operation:** execute set operations (e.g., union, intersection) on sets of vertices.

As mentioned, the combination of these low-level tasks enables the achievement of more complex tasks, which, in the multivariate network context, have been classified as structure-based, attribute-based, estimation, or browsing tasks [20].

1.1.2 Information Visualisation

Although the meaning of information visualisation has changed over time, today, this term indicates the visual, computer-supported presentation of abstract data where people are empowered to interact, via direct manipulation, with such information to expand their understanding about that specific domain under study [26]. It is has become an essential tool supporting science and decision-making processes because it, if presented properly, enables the rapid interpretation of big data [27]. To create two-dimensional or three-dimensional layouts representing information is already a challenging activity on its own. Nonetheless, the complexity degree of such activity becomes even more substantial when the represented data has many dimensions (or attributes), as in multivariate networks. By acknowledging the existence of such difficulties, various graph drawing methods have been developed to illustrate the vertices and edges of a graph, each producing a distinct network representation, such as node-link or matrix-based diagrams, which work well for a small amount of simple data. However, even with many existing techniques for combining such network visualisations with multidimensional data, such as colour-coding and labelling, the scalability of this process is limited with clusters quickly emerging, which means that these techniques do not entirely solve the visualisation issue for real-world data sets [20]. In his PhD thesis, Jusufi identified different visualisation approaches that mitigate the multivariate network visualisation problem [17].

A second component of the information visualisation field, equally important as the graphical network representation, concerns the interaction between users (e.g., InfoVis researchers, data analysts) and visualisation systems, as the former explore and perform analytic tasks on data sets to extract hidden but valuable information and relationships [28, 29, 30]. However, the current nomenclature systems used for identifying and classifying such interactive tasks do not necessarily converge, as there are different granularities and manners to describe the techniques [22]. In addition to the multivariate network tasks previously described, some view-level interaction techniques are also relevant in the context of this project; they provide to the users the means for navigating through the network and focusing on different elements of interest. The following list outlines the highlighting and navigation actions according to the research work of Wybrow *et al.* [31]:

- **Highlighting:** this category includes hovering, brushing and linking, and magic lenses techniques. Visualisation systems usually support such actions when they concurrently display the same data set in different but linked graphical views. The implementation of hovering and brushing and linking further support the effectiveness of multiple graphical representations, as it enables the emphasized observation of an entity in all views when the mouse is moved over or hovered over the same element in one layout. Magic lenses, on the other hand, enable users to focus on entities even in dense areas of the network by changing the graphical exhibition of such elements;
- **Navigation:** this category includes panning and zooming and view distortion techniques. Panning and zooming are actions that enable users to adjust the displaying viewport to reach and visualise the network areas of interest. There are several manners in which these actions are supported, varying from different hardware technologies, such as mouse wheels, to actual software design decisions. The view distortion enables users to better inspect entities of interest by adding extra space to such elements. Although fisheye is a popular distorted view, specific distortion techniques can be applied to edges and nodes, such as Edge Lenses and Balloon Focus, respectively.

1.1.3 Human-Computer Interaction

The interdisciplinary field of human-computer interaction (HCI), which involves human factors engineering, computer science, and cognitive science, has become a topic of great interest to academic researchers as they investigate the diverse manners of communication between users and computers. Although the HCI domain shares similarities with other areas, such as user experience (UX) design, it mainly focuses on academic discoveries with foundations on the experimental understandings of users [32]. After years of research work and advancements in technology, today, humans are able to interact with diverse systems through other interfaces than the standard mouse device, including touchscreen, hands gesturing, eye tracking, voice and face recognition, and brain-machine interfaces. All these interfaces belong to the Natural User Interface (NUI) category, as they endeavour at providing natural, effortless and invisible ways of communication to end-users. Büschel *et al.* [33] started a discussion concerning the reasons and manners in which interaction techniques that belong to the NUI category in association with archetypal hardware setups can be utilised to support immersive environments for data analytics, strengthening user engagement and possibly improving efficiency. From the various interface alternatives, this study focuses on the hand gesturing one.

Computer gestures recognition is a multifaceted process that involves activities from the creation of a gesture mental model and its mechanical execution, over the extraction, modelling and analysis of hand features and movements, to the mapping of patterns and eventual application of machine learning techniques. During this process, diverse factors can affect the overall execution performance of an HCI task, including human aspects, such as levels of comfort and possible motor restrictions imposed by the use of input hardware, and also computer aspects, such as image capture and computational power [4]. Figure 1.3 illustrates the gesture recognition process, and how such external factors relate to the process activities.

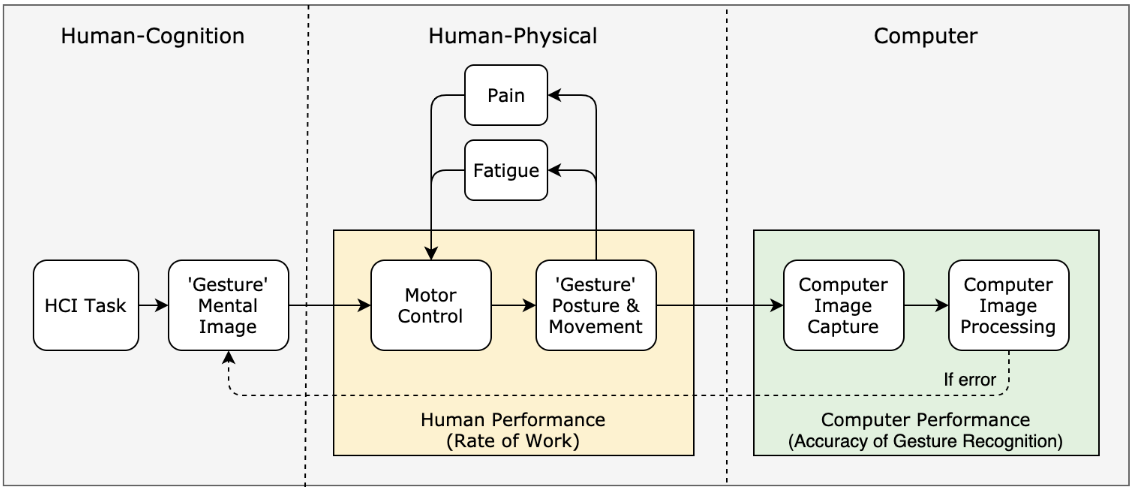


Figure 1.3: This image shows the model described by Rempel *et al.* in [4]. It illustrates the computer gesture recognition process, and the manner in which human and computer aspects, such as pain, fatigue and computational power, relates to the overall performance achieved in the execution of an HCI task. The process is divided into three phases (human-cognition, human-physical, and computer), and the activity flow starts with the creation of a gesture mental model followed by the motor execution of such a model to the conclusion of the flow with the computer image recognition and information processing. It is also important to notice the backward loop path to the initial phase of the process in case of error in the computer image processing step.

Although there are distinct taxonomies categorising gestures according to their non-physical characteristics, the system designed by Quek, summarised in Figure 1.4, suits well the HCI context of this project [5, 6]. The purpose of communicative gestures is to transmit information through visual interpretation of hand elements, where fingers and palm movements have meaning to someone. On the other hand, the use of manipulative gestures is more relevant in the context of interaction with objects. Moreover, strategies for gestures recognition also have classifications; according to Murthy *et al.*, they are divided into rule-based and machine learning-based approaches [34]. In the first category, hand features are juxtaposed against implemented rules while, in the latter, gestures are regarded as the outcome of a stochastic process.

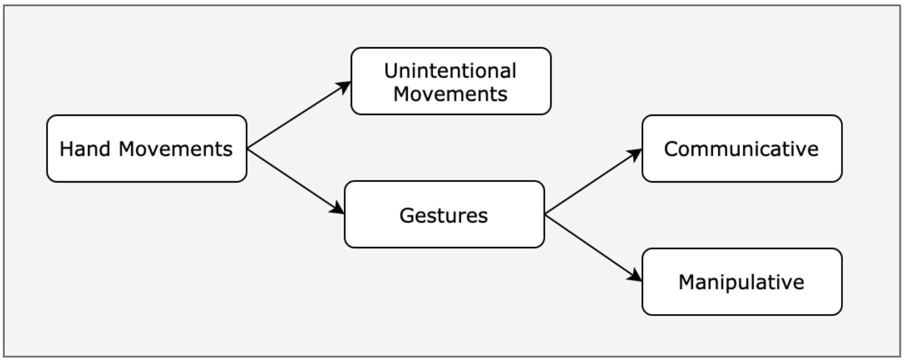


Figure 1.4: This image illustrates a condensed version of the gesture taxonomy introduced by Quek in [5, 6]. According to this taxonomy, hand movements are first categorised either as unintentional movements or gestures. The latter indicates that a purpose and intention exist in the movements executed by an user when interacting with a system while the former refers to the movements that result from transactions between gesture positions and even natural human reflexes. Then, gestures are further broken down into communicative and manipulative gestures, according to their purpose and characteristics.

Another significant viewpoint to be analysed when approaching the gesture interface domain concerns the actual designing of gestures. The definition and identification of the most comfortable, natural, rememberable, effortless or invisible gestures have not yet converged to a well-defined set. Over recent years, researchers in several domains where Augmented Reality (AR) is inserted (e.g., entertainment and medicine) have proposed and analysed distinct sets of gesture designs. Such gestures do not always overlap, as it is a complex task to achieve and compile a unique set of gestures that is the best for all purposes. The notion of good or bad intrinsically depends on the domain of application, executed tasks and also employed technologies. Nonetheless, in a comprehensive study, Piumsomboon *et al.*, from empirical observation, elicited user-centred gestures for forty tasks common to applications employing Augmented Reality [7]. Figure 1.5 identifies such tasks and their correspondent gestures.

1.1.4 Leap Motion Controller

The Leap Motion Controller is an input technology which, by tracking hand elements, enables humans to interact with computer machines through gestural interfaces. The controller hardware is manufactured as a small module device that peripherally connects to a computer. The device contains three strategically positioned infrared LEDs that prevent overlapping and two built-in high-resolution cameras that work as image sensors [35, 36]. The controller device creates a three-dimensional interactive field of approximately 60 centimetres, where the hand tracking software virtually captures hand and finger gestures. Complex and subtle movements of joints and bones, which are often more difficult to dis-

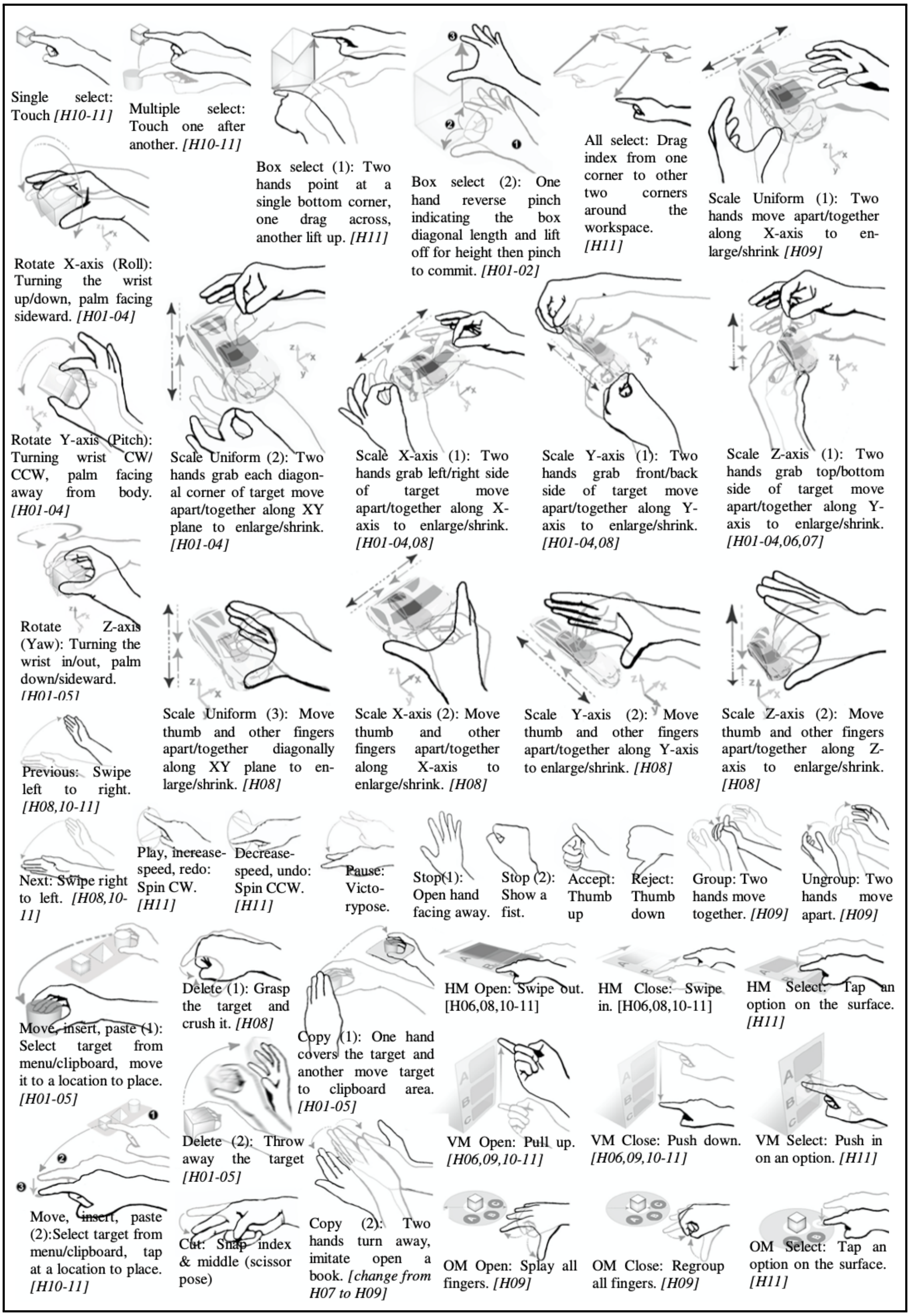


Figure 1.5: This image identifies 40 tasks that are commonly found in AR applications, and their correspondent user-centred hand gestures. This figure is retrieved from the paper written by Piumsomboon *et al.* [7].

tinguish, are also detected [37]. Figure 1.6 shows the controller device and illustrates the virtual model created by the hand tracking software.

Figure 1.6: This figure presents the Leap Motion Controller device and illustrates the human-computer interaction style supported by the technology. The image was retrieved from the Ultraleap website [8].

The enterprise currently responsible for such a technology, namely Ultraleap, claims that it optimises human interaction to digital worlds, taking it up to an effortless and natural sensation [8]. With infrared LEDs pulsing and sensors feeding data into the software more than 100 times per second, the technology delivers a tracking system that achieves almost-zero delay and exceptional accuracy [38]. Real-world problems in several areas including not only entertainment but also medicine and healthcare, personnel training, manufacturing and household, already experience the benefits that gesture user interfaces can offer [35, 37]. For example, the use of the leap motion controller is convenient for maintaining sterile conditions during dental surgery procedures as it enables dentists to touchless navigate through images [39].

1.2 Problem Formulation

Today, there is no published study testing the series of usability improvements which presumably can be observed in the interactive analysis of two-dimensionally displayed multivariate networks due to a shift in the current standard mean of interaction, the mouse device, to an interface based on hand gesture recognition. It remains unknown the effects that such a move in the human-computer interaction style could cause to the overall user experience and execution performance of tasks conducted during the analyses of such data structures. However, it is not only research that is lacking; no actual network visualisation software has been comprehensively and thoroughly integrated with the Leap Motion Controller technology. There is no professional visualisation software system employing such a tool as the standard mean of interaction with multivariate networks. The milestones planned for this degree project share the same purposes of enabling and guiding a research work concerning the use of hand gesture recognition for the interactive analysis of such multivariate networks. The gradual achievement of these project milestones provides the answers for the following research question.

Research Question: Does an interface based on hand gesture recognition provide a more pleasant and effective user experience in comparison to standard mouse devices for the interactive analysis of two-dimensionally displayed multivariate networks?

It is relevant to mention that effectiveness, in the context of the research question, is regarded in terms of the accomplishment of a task (i.e. the required effort for executing a task to its completion). Besides, throughout the investigation of the research problem, related questions are also addressed, such as:

- What combination of hand gestures among the available options feels the most intuitive and natural for the users to perform certain tasks?
- What kinds of tasks are most likely to produce better results in terms of usability when considering such a change in the means of interaction?

1.3 Motivation

Network structures are growing more and more complex as real-world data sets become more and more comprehensive but represented and interpreted nevertheless. In interdisciplinary domains where the application of multivariate networks is essential for modelling relational data, such as biochemistry, social network or software engineering, appropriate data illustration and clustering are recurrent challenges for researchers [18]. To overcome such difficulties, the focus of recent research on information visualisation, according to Yi *et al.* and Lee *et al.*, has been mostly dedicated to the representation facet of visualisation software systems rather than to the interaction aspect [28, 22]. However, some factors, including the provision of facilities for interacting with data, can also play a major role in mitigating such problems [20]. Most of the tasks performed by researchers and data analysts when investigating multivariate networks have already been determined; however, the standard mean of accomplishing them is through mouse devices, which can be cumbersome and restrictive at times. The integration of touchless interfaces with visualisation software might be positive and extend the possibilities within the multivariate network field. An interface based on the recognition of hand gestures might improve the overall user engagement and experience in the interactive analysis of complex network structures as well as increase task execution performance, taking us one step closer to solving the existing graph visualisation problems.

1.4 Objectives

During the initial discussion phase with the project supervisor and colleagues, specific goals were identified and established as the essential deliverables of this degree project. To be able to obtain empirical data and answer the question previously introduced in the problem formulation section, whether the interactive analysis of multivariate networks through hand gestures offers more benefits in comparison to standard mouse devices, a platform that can track and recognise hand movements enabling in this manner analysts to execute their tasks is required. Therefore, the first objective of this project, which is an infrastructure for the accomplishment of the main goal, is to design and implement a web application which integrates existing network visualisation tools with the leap motion controller technology.

Objective 1: Build a web application that enables the interactive analysis of two-dimensionally displayed multivariate networks through hand gestures

An extra functionality of the system that is worth mentioning is that the application enables analysts to upload graph files to the system as an alternative for generating random networks. For the execution and development phase, this first objective needs to be broken down into low-level milestones to guide the planning of the work, as the following list shows:

O1.01 Web application integrated with simple network visualisation
O1.02 Literature on multivariate network tasks reviewed
O1.03 Multivariate network tasks to be implemented selected
O1.04 Functionality for multivariate network tasks implemented
O1.05 Leap Motion Controller integrated into the web application
O1.06 System tested and source code refactored
O1.07 Hand gestures selected and recognition implementated
O1.08 System tested and source code refactored
O1.09 Multivariate network tasks linked to the hand gestures
O1.10 System tested and source code refactored

However, it is essential to employ gesticulations that feel intuitive and natural to enable analysts to work with ease. Since human-computer interaction plays a major role in this application and the milestone O1.07 requires a series of hand movements to be defined, it is inherent that a second relevant objective of the project must be a conceptual study on hand gestures for interaction with software systems.

Objective 2: Perform a conceptual study on hand gestures and establish their mapping onto multivariate network tasks

The two milestones that lead to the fulfilment of this objective are:

O2.01 Literature on human-computer interaction and hand gestures reviewed
O2.02 Mapping of gesticulations onto multivariate network tasks defined

Subsequent to the completion of the two first objectives, the concluding goal comprises the collection and analysis of empirical data to answer the proposed problem. In order for data to be collected, it is required the preparation of one or several environments (network topologies) and the definition of tasks that the subject(s) must perform Thereby, the last objective of this project is to examine the overall user experience and performance of analysing multivariate networks through standard mouse devices against hand gestures interactions.

Objective 3: Compare the overall user experience of standard mouse devices against hand gestures guided interactions for the interactive analysing of two-dimensionally displayed multivariate networks

There are concrete actions required for achieving Objective 3 that should be carefully planned, as the completion of the objective involves and depends on external people who do not have a direct interest in the success of the project. The set of milestones that constitutes this objective is:

O3.01 Participant(s) contacted for the collection of empirical data

O3.02 Empirical data collected

O3.03 Data visualised (charts, diagrams) and analysed

It is important to notice that objectives 1 and 2 constitute the infrastructure required for the achievement of objective 3. It explains the dependencies between some of the milestones and the importance of following the defined project plan. From the initial discussions until the analysis of the empirical data, the expected result for this study had been that the use of hand gestures for the interactive analysis of multivariate networks would improve the overall user experience and execution performance of some tasks while standard mouse devices would still remain more suitable for other tasks.

1.5 Scope/Limitation

As aforementioned, the scope of this project entails three relevant deliverables; the first one being the web application that enables the interactive analysis of multivariate networks through hand gestures. However, due to time and knowledge constraints, restrictions on the system requirements are unavoidable, and they affect how comprehensive the software is. The limitations are:

- Although there are software systems that support multivariate network analysis and visualisation, this application enables users to work with two-dimensional network representations only;
- Although the literature is extensive and several tasks can be identified for analysing multivariate networks, only the most common or well-known tasks are taken into consideration for the development of this system;
- Although there are software systems that implement algorithms and connect with graph databases to support extensive networks, this application's performance is not optimised to handle graphs containing thousands of nodes or edges.

The second deliverable defined in this project's scope is the study on human-computer interactions, primarily focusing on hand gestures to identify the most suitable movements for interaction. Since the literature is extensive and time is limited, restrictions also apply to this study. The limitations are:

- Although it is socially meaningful to develop systems which include and enable everyone to use and benefit of technology, accessibility will not be included as a quality attribute of the application, meaning that the hand gestures to be implemented might not be suitable or ideal for disabled people;
- Although experiments would help to decide which hand gestures or combination of actions are indeed the most suitable for each task, this decision is supported mostly by the literature. The exception circumstance is if two or more gesticulations seem appropriate for a task, then a brief experiment is conducted.

Finally, the third deliverable entailed in the scope of this project is the empirical activity comparing, from a high-level perspective, the system usability achieved by the proposed interface against conventional mouses devices for the interactive analysis of multivariate networks. The conclusion of the project has its foundations on the data set accumulated throughout this phase. However, a limitation of great proportion affected both the research population involved and the method used. Due to the restrictions and recommendations imposed on the societal interaction behaviour by the outbreak of novel coronavirus, which happened exactly during the execution of this project, it became health-wise dangerous and also not feasible to engage, interview, and conduct experiments with individuals of a population for the collection of experimental data. Although the initial intention was to conduct a study case involving several experiment subjects to assess and evaluate the developed system by measuring execution performances, the impacts of such restrictions on this project are reflected both by the change of method chosen for the evaluation of the interface, which is further described in the third chapter, and the limited number of participants. The diverse constraints narrowing the extent of this project may insert bias and uncertainties into the result and conclusion of the study.

1.6 Target Group

Several people, mainly categorised into groups according to the work activities they exercise and their stake in the success of this study, might be interested in reading this paper. Naturally, the different target groups have distinct levels of curiosity and engagement, depending on their scientific background and role in the project. Table 1.1 identifies the different project target audiences and their motivation.

1.7 Outline

Heretofore, a thorough explanation of the study proposed in this degree project has been provided, including not only the problem, motivation and objectives behind the research work but also the background knowledge required to understanding the scientific grounds that support this project. The remaining work comprising this thesis is developed throughout seven subsequent chapters, as briefly outlined in this section. The second chapter introduces the readers to a couple of studies that share related purposes in the same fields of study as this thesis, including the presentation of their results and conclusions. The third chapter describes the scientific problem-solving activities employed throughout the project life-cycle to resolve the previously introduced research question. It also includes a discussion concerning the reliability and validity of the project as well as ethical considerations which require attention. The fourth chapter contains an architectural description of the developed web application, including requirements, implementation, testing and deployment details. Moreover, it includes a discussion on the choices of technologies (e.g., libraries and frameworks) considered throughout the development phase. The fifth

Audience	**Motivation**
Project supervisor	Prof. Dr. Andreas Kerren has a distinctive interest in this study, from the choices of technologies implemented to the outcome. Initially, the project was his proposal, and it can guide the focus or benefit future research work carried out by the information visualisation group at Linnaeus University
Thesis examiner	As someone of established competence in the domain of information visualisation, the thesis examiner is interested in this study to stay updated with the latest relevant information; and also because the assessment and evaluation of this paper is his/her responsibility
InfoVis researchers	Information visualisation researchers have an interest in common with the thesis examiner. In academia and research environment, it is fundamental to stay updated with the latest discoveries in the field of study; they can become a source of thoughts and shift the direction of future work
Data analysits	Professional data analysts who also handle data visualisation might have an interest, to a medium extent, in the results of this investigation. Businesses are regularly striving to increase effectiveness and productiveness; therefore, the results of this paper might directly affect how analysts work
Scientific community	Since the visualisation and analysis of multivariate networks apply to several domains of science, the search for innovative approaches that result in performance improvement of data analysis might interest, to a medium extent, the scientists overall, as the outcome obtained can benefit their work
Ultraleap	The enterprise responsible for the Leap Motion Controller might have a small interest in scientific papers to identify in which fields the technology is being employed. It can lead to enhancements in the system to better support these domains in future

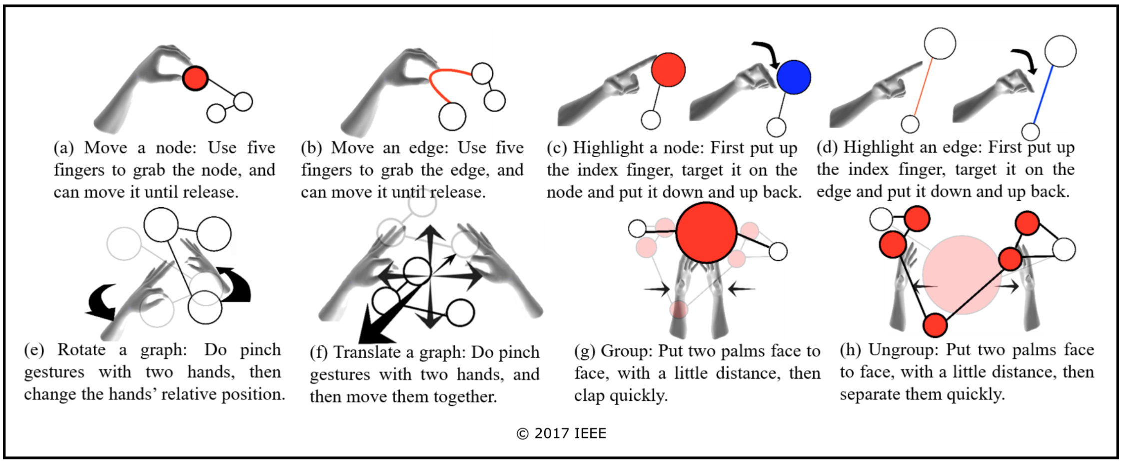
Table 1.1: Project's target audiences and their interests

chapter describes the developed interfaces—both designed hand gestures and mouse interactions that are mapped to the implemented data analysis tasks—by making use of both text and illustrations to help the readers better understand such interactions. The sixth chapter describes with a great level of detail the concrete settings for expert review activity as well as the feedback provided by the expert after testing the proposed gestural interactions. The seventh chapter interprets the results and analyses the findings and their relationship with related works. Ultimately, the closing chapter concludes the research work with a summary of the study, describing its relevance and applicability for science, society, and companies. Moreover, it includes a prospect for future work.

2 Related Work

Although this research requires knowledge on graph algorithms and visualisation tools in the context of multivariate networks, the scientific and innovative value offered correlates mostly to the field of human-computer interaction. This project explores the design and deployment of a three-dimensional gesture interface, supported by the Leap Motion technology, and the potential usability and performance improvements that a shift to such interface can prompt in the interactive analysis of two-dimensional multivariate networks activity. Similar studies have been conducted in the same field as this work, but with variations in the leading technologies or essence of the networks. Notwithstanding the discrepancies, such studies are still valuable sources of knowledge where one can obtain information not only about hand gestures design, as introduced in the previous chapter, but also about the potential outcomes that the particular specifications of this project can produce.

Until the present moment of this writing, there is no identical research comparing, specifically for (1) two-dimensionally displayed multivariate networks and (2) interaction interface deployed via Leap Motion Controller technology, the use of hand gestures as a form of interaction input against standard mouse devices. However, researchers have examined both input approaches in the manipulation and analysis of other graphs. Huang *et al.* [9] performed a similar study using the Leap Motion technology for common graph operations in virtual reality (VR) environments; the authors propose a set of operations for different graphs (Force-directed [40], Brain [41], and BioLayout [42] graphs), including finding adjacent nodes, finding the shortest path between two nodes, and counting all nodes with a determined property. The obtained outcome shows that participants achieved most of their tasks with higher performance and accuracy when using the gesture interface in comparison to the mouse pointer. Moreover, it also indicates that users were reasonably comfortable with the set of designed gestures, illustrated in Figure 2.7.



(a) Move a node: Use five fingers to grab the node, and can move it until release.

(b) Move an edge: Use five fingers to grab the edge, and can move it until release.

(c) Highlight a node: First put up the index finger, target it on the node and put it down and up back.

(d) Highlight an edge: First put up the index finger, target it on the edge and put it down and up back.

(e) Rotate a graph: Do pinch gestures with two hands, then change the hands' relative position.

(f) Translate a graph: Do pinch gestures with two hands, and then move them together.

(g) Group: Put two palms face to face, with a little distance, then clap quickly.

(h) Ungroup: Put two palms face to face, with a little distance, then separate them quickly.

Figure 2.7: This image depicts a set of graph operations and the corresponding set of hand gestures to conduct each task. The gestures illustrated in this figure were designed focusing on VR environments. It is fetched from the paper written by Huang *et al.* [9].

Another project, not so structured and scientifically supported as the previous one but still strongly related to this study, proposes the use of gesture interface for manipulation and analysis of the Twitter network. Burshtein *et al.* developed a visualisation application that enables users to explore the particular multivariate network, also using the Leap Motion controller as input and navigation device, from a three-dimensional view [10]. In the application, navigation and highlighting tasks are implemented, such as zooming in and out, panning, expanding and collapsing data, and rotating viewpoint, as exhibited in the project's demo video [43]. The authors of the project could not conclude whether the use of gesture inputs improved the user experience due to incomplete, inaccurate, or limited implementation. Nevertheless, the project still provides insights into the environment and technologies employed and how they are integrated as well as the set of hand gestures utilised. Figure 2.8 illustrates their project.

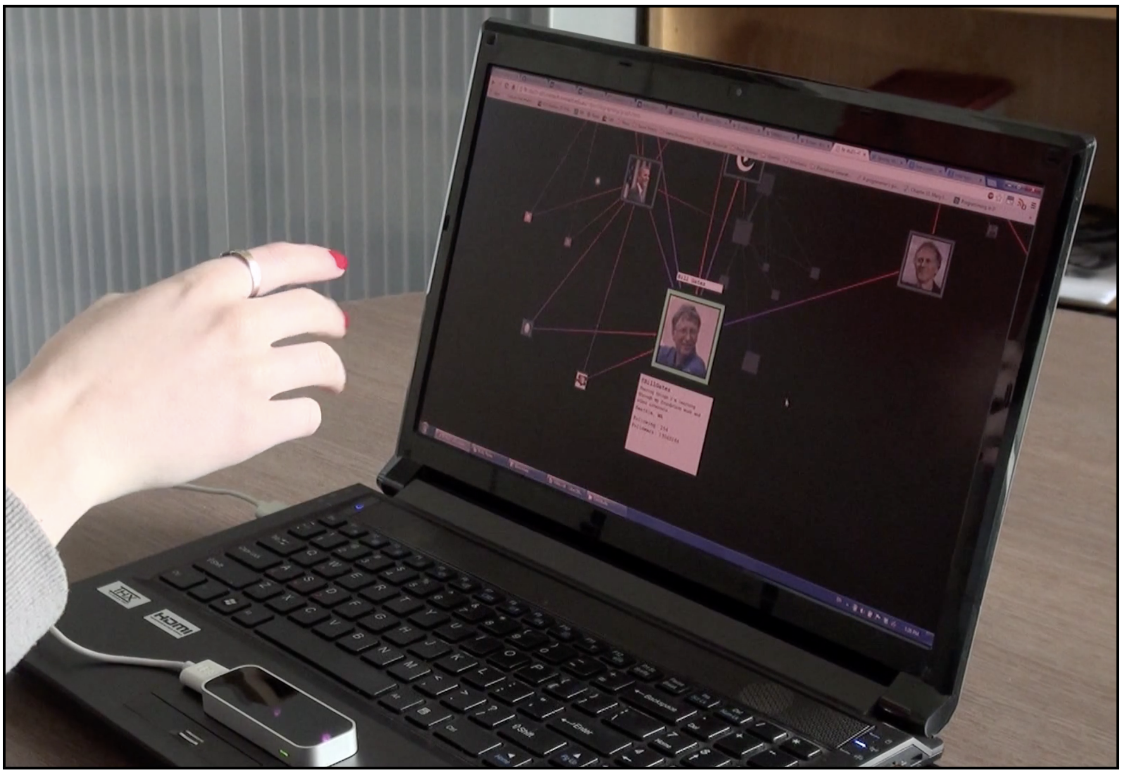


Figure 2.8: This image shows the manner in which the application developed by Burshtein *et al.* relates to this project; it uses the Leap Motion Controller for the interactive analysis of a MVN. However, their project was specifically developed for the Twitter network, and uses three-dimensional perspective to represent it. This figure is a screenshot of the project's demonstrational video [10].

3 Method

A combination of qualitative and quantitative methods are conducted to answer whether gesture interfaces offer valuable advantages for the interactive analysis of multivariate networks in comparison to standard mouse devices. A method, in this context, refers to a problem-solving activity that provides an organised and structured manner of approaching and addressing that problem [44]. The proposed research question searches beyond a simplistic yes or no answer; it explores the developed interface and investigates the relationship between the implemented hand gestures and the overall user experience. As a sole method would not be enough for creating the required infrastructure, developing the interface, and questioning the usefulness of it, this project comprises the following verification and validation, literature review, and expert review methods to tackle each of the project objectives, as shown in Figure 3.9. Notice that the methods were carried out following an iterative approach, as illustrated by the circular arrow.

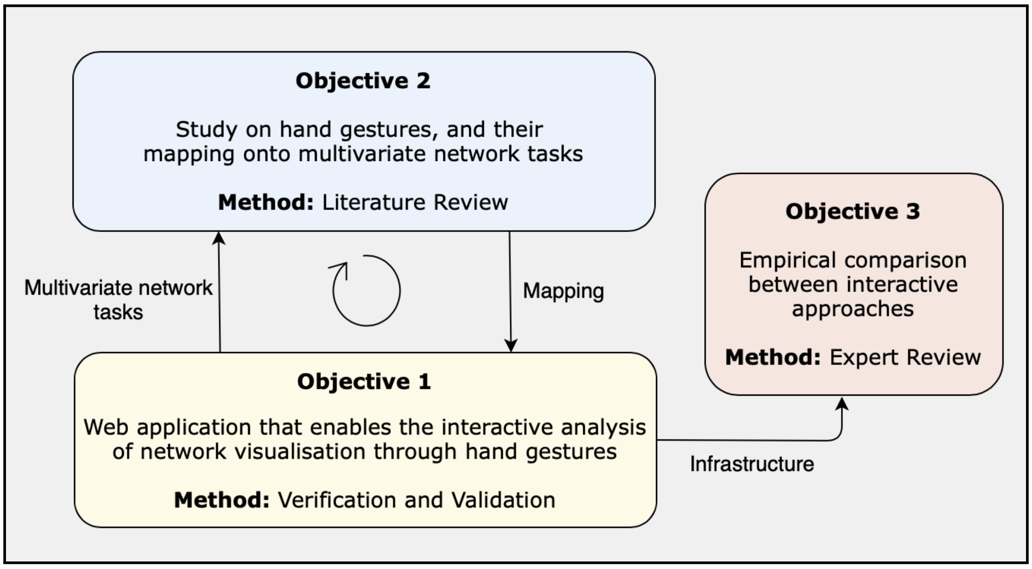


Figure 3.9: Dependencies between objectives.

3.1 Verification and Validation

The verification and validation method comprises different techniques that are utilised to verify whether the concerns, formally known as functional and non-functional requirements, presented by the stakeholders of a project are met by the software system under development and deliver value to its clients [45]. Although there are several definitions of verification and validation in the literature, the explanation presented by Bahill *et al.* suits this subsection very well, as it is abstract, concise and clear: system's verification guides the development team to build the system right, while system's validation helps the development team to build the right system [46]. From this explanation, it is possible to understand the verification process as a low-level activity which tests system requirements, and the validation as a high-level activity which tests the system as a whole against the customer or end-users expectations.

The verification process was iterative throughout the implementation phase. Although the set of requirements for the application were not extensive nor extremely precise, as described in the next chapter, exploratory and manual test cases were regularly conducted to check the new features implemented in each iteration. This testing approach is further detailed in Section 4.4. The validation phase was also iterative and incremental. During the initial stages of the development process, the project supervisor was not consulted with demanding frequency. However, as the project advanced towards its completion, regular meetings, both in-person and online, were scheduled with the supervisor where he had the opportunity to provide valuable feedback expressing his expectations for the application. By employing the verification and validation method throughout the entire development life-cycle, the authors gained knowledge and confidence in the created system.

3.2 Literature Review

Although a systematic review awards higher scientific value to research in comparison to a critical literature review, it requires considerably more resources, such as time and effort [47]. Taking such fact into consideration, and understanding the main goal of this work, which is not to summarise with a great level of depth and details the existing knowledge in a precise topic, the traditional approach of review literature was preferred. The reason for carrying out a literature review throughout the entire project life-cycle was to constantly obtain insights into the different domains included in this study, specifically multivariate networks, information visualisation, and gesture interface. At the beginning of the project, the focus of the literature review was to extract general knowledge about the involved subjects as a whole. However, as the project advanced through its phases, it was necessary to perform a more thorough review of each subject. It is also worth mentioning that additional attention was devoted to reviewing the core point of the project: the gesture interface. It was crucial to understand the elements that compose a gesture interface, from the design of hand gestures over to the system required to deploy it to the user experience. The literature review mostly consolidated knowledge and information from peer-reviewed articles and books but did not exclude other sources, such as similar projects, technology blogs, technical reports, and videos.

3.3 Expert Review

The expert review method, as the self-explanatory name discloses, relies on people with expertise on a particular research field who can evaluate and assess with precision the outcomes of a study, which can be both a scientific paper or a software system. In this reviewing activity, as indicated by Yue *et al.*, it is of absolute importance that experts have, ideally, no competing interests or association with the author(s) of the project and enough technical knowledge and research practical experience with the research topic under study to ensure fairness, impartiality and accuracy of the process [48]. In the case of this project, the product under evaluation is the developed web-based application, particularly its gesture interface. The expert conducts a comparison of types of interaction—mouse-based versus gesture recognition interfaces—with the guidance of a questionnaire for the evaluation process. Also, the appointed expert meets the aforementioned expertise criteria. The remainder of this section provides a generic description of the method, like advantages, drawbacks and applicability. Nevertheless, Chapter 6 provides supplementary information describing the specifics of the expert review method in this thesis project, such as what expert evaluates, the reasons and the manner in which the evaluation is conducted.

Inherently, the selection of this method as an assessment tool introduces difficulties to the evaluation process, including the required resources (e.g., time and money) to find sufficient experts and the need for recording the experts' perceptions. According to the U.S. Department of Health & Human Services, however, the benefits that this method offers are: (1) products can be easily and quickly evaluated even on early stages of the design process; (2) it produces less formal reviews for complex systems; and (3) it can be combined with other usability testing methodologies to reveal other potential issues [49]. Moreover, this method can be understood as an alternative to the case study method; researchers often encounter challenging degrees of complexity in elaborated experiments as controlling and measuring all variables that interfere with the performance of such experiments is difficult.

In the field of information visualisation, according to Tory *et al.*, the expert review activity aggregates relevant feedback to the evaluation outcome of visualisation systems [50]. Nonetheless, they conclude that this method should not entirely substitute user studies, because experts are not always able to identify all usability problems. Therefore, the expert review method is utilised to produce the pilot data required for the analysis and consolidation of conclusions about the system proposed in this project.

3.4 Reliability and Validity

The choice of method(s) for evaluating a software system or the content of scientific papers and the manner in which data is collected can introduce reliability and validity issues into research findings. Although researchers and people in academia are often aware of such threats, it not always possible to completely mitigate them. In this project, as previously reported, the chosen method for such evaluation purpose was the expert review in conjunction with a questionnaire. Following are the reliability and validity threats identified due to both the unavoidable human essence of the method itself as well as the limitations imposed on the project scope.

The reliability of this project refers to its accuracy and reproducibility; it indicates the likelihood of the results obtained repeat themselves in case the research is reproduced. Taking into consideration the subjectiveness implicit in the expert review activity and in the object of study—the overall user experience of the gesture interface developed—it is possible to ascertain, according to Wilson [51], that the results manifested in this paper are not the most reliable. Moreover, there is an aggravation of the reliability problem, as supported by Babbie [52], since only one person with expertise in the domains of human-computer interaction and network visualisation was consulted. Therefore, any human reaction that may affect the expert during the reviewing process (e.g., fatigue, hunger, and personal problems) can potentially alter the perceptions of the expert about the system. Nevertheless, the study tries to minimize this subjective aspect of the process by introducing a questionnaire whose purpose is to guide the expert objectively. Furthermore, the lack of a well-consolidated and structured hand gestures system compromises the reliability of this study. The determination of a set containing the hand gestures recognised as the most intuitive and effortless for the interactive analysis of multivariate networks depends on the perception of the authors throughout the development of the project, and, therefore, may change over time as additional literature sources are reviewed.

The validity of this project refers to the degree at which the research findings provided are correct (or valid) and truly express the real world. Although there exist various forms of research validity threats, the most critical ones are construct validity, internal validity, and external validity problems [53]. The internal validity threats in this project mostly relate to (1) the complexity of controlling all variables that may affect the expert review activity, and (2) to the fact that the expert is an associated member of the project supervisor's research group, which can contribute to a biased result. Subsequently, the external validity issue is produced by the difficulty in accurately generalising the experience of one expert to the entire community of graph visualisation researchers and data analysts. Lastly, the construct validity concern in the evaluation of the gesture interface is due to the high-level nature of expert review activities in comparison to low-level and measured case studies focused on performance. Also, the authors' lack of professional expertise in software development concomitantly with the use of manual and exploratory testing and restricted time may affect the implementation of the gesture interface and the application structure itself, which possibly compromise the validity of the project. Nevertheless, following the recommendations of Cohen *et al.* [54], the authors tackle the qualitative data validity issue by objectively and disinterestedly approaching the research work and by maximising, within the constraints imposed, the depth, fairness, richness, and extent of the data obtained.

3.5 Ethical Considerations

The intention and design of an experiment or evaluation plan should take into account ethical issues that are associated with such a plan. The types and extension of ethical considerations that should be addressed in a research work depend greatly on its topic and methods employed. Although the ethical concerns raised by Tessier *et al.* [55] apply to experiments with participants, some of them also relate to expert review method utilised in this project. The following list identifies such concerns and describes suitable solutions put in place for mitigating them.

- *Comprehensive project information*: the expert was thoroughly informed about all the aspects and final objective of this project before the reviewing process.
- *Freedom*: as a professor/PhD student relationship between the project supervisor and the expert was noticed, the latter was explicitly informed that the cooperation in this project was entirely optional.
- *Privacy and confidentiality*: the expert was informed that his/her identity could be removed from the study and not exposed beyond the project team if desired.

4 Software System

The development of an infrastructure that enables, through the use of hand gestures, the interactive analysis of multivariate networks is the first objective of this project. The outcome is a software system where users can perform some of the recurrently identified tasks in the analysis of multivariate networks using a touchless, hand gesture recognition interface supported by the Leap Motion technology.

4.1 Requirements

During the discussion phase with the project supervisor (process formally referred to as requirement elicitation according to the engineering terminology), some requirements, which the software system must comply with, were identified. The first one concerned the tracking of hand elements for the recognition of gestures; it must be accomplished by utilising the Leap Motion Controller device as the mean of technology. Subsequently, the second requirement contemplated the nature of the system; it must be a web application accessible from the major browsers: Chrome, Safari and Mozilla Firefox. Finally, the last requirement concerned the language of implementation; it has to be JavaScript.

The focus of this research is not to deliver the most sophisticated network visualisation software, but to study how the proposed interface affects the performance of multivariate network analysis. Thus, taking it into consideration together with the available resources, it is understandable that no requirements regarding software performance or security were determined. Today, several architectural tactics are employed in network visualisation web tools to improve efficiency, such as performing computationally expensive processes on a server or using graph-based databases (e.g., Neo4j) for the optimisation of queries. Such techniques enable the analysis of larger networks with less rendering delays. As explained, however, the software system is implemented in a simple but structured manner given the purpose of the work.

Despite the flexibility regarding quality attributes allowed by the customer, who in this context is the thesis supervisor, there is a natural and implicit expectation for the software system to graphically resemble, at least to a small degree, other multivariate network analysis tools available on the web. Considering the node-link diagram type visualisation, the software must ease the observation of results from analytical tasks by rendering nodes and edges in different colours upon accomplishment of actions (e.g., highlight nodes captured in a selection or after the computation of the shortest path). Moreover, the software must also enable users to navigate through networks by using panning and zooming actions.

Although view-level interaction is fundamental in graph visualisation applications, the analysis of multivariate network also requires other analytical tasks to be made available by software. Taking into consideration the study case desgined in this research and the assignments which it contains, the system here described must enable users to select and deselect nodes and edges; users must be able to select single or multiple entities by individually choosing one after another or by creating a selection area. Also, the system must deliver functionality to find adjacent nodes to a selected vertice(s), determine the shortest path between two vertices, and discover connected components upon request. Moreover, users must be able to revisit previous selections and filter entities by attributes. All the described functionalities must be accessible both by using standard mouse devices and hands gestures for performance comparison. The following list summaries the elicited technical requirements (TR):

TR 1: the touchless hand gesture recognition interface must be deployed through the Leap Motion Controller device;

TR 2: the software system must be delivered as a web-based application;

TR 3: the system must be compatible with top mainstream browsers;

TR 4: the language of implementation must be JavaScript;

TR 5: users must be able to navigate through networks by zooming and panning;

TR 6: the application must be capable of rendering edges and vertices in different colours, with the possibility of displaying text labels on each vertice; and

TR 7: the system must make available, both through mouse and hand gestures, analytical functionalities that enable users to accomplish the assignments presented in the study case.

4.2 Technologies

A simple single-page application (SPA) can be implemented by writing code only in plain JavaScript, also known as Vanilla JavaScript. However, in a professional setting where systems are expected to comprise complex functionalities and interactive user interfaces, this development approach does not remain optimal. It would require several hours of work and many hundred lines of code to achieve such requirements. For such reasons, JavaScript frameworks and libraries were utilised for the development of this system. The reasoning behind each of the decisions regarding choices of technology is discussed in this subsection. Figure 4.10 provides the design space in shape of a decision tree.

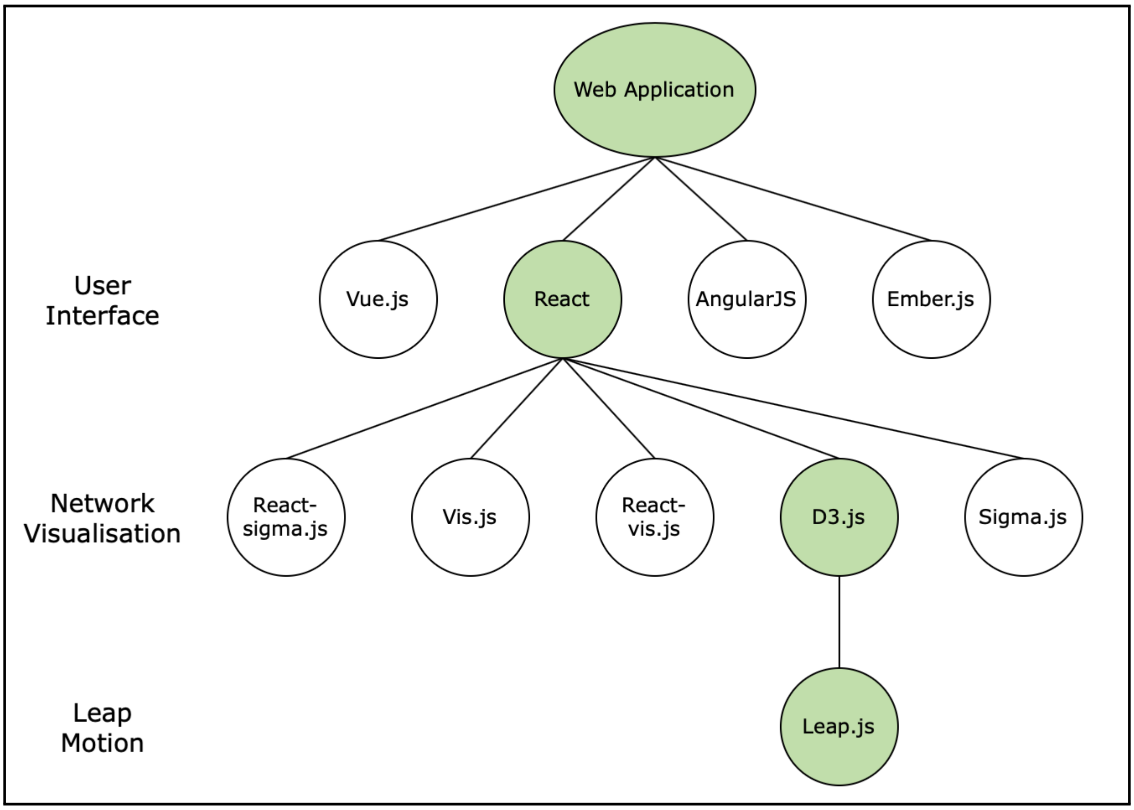


Figure 4.10: This diagram identifies the choices of technology through a design space decision tree. It shows the different technologies taken into consideration when implementing the application and the path taken.

4.2.1 Front-end Framework

There are several JavaScript frameworks and libraries for front-end development that ease developers' tasks of creating dynamic user interfaces. Although the specific details of such technologies vary, their major purpose often is to enable the production of instant feedback (response) as guests interact with web applications. Up-to-date toolsets are also associated with code maintainability and efficiency improvements. However, there can be some shortcomings in relying on a front-end framework and not being responsible for the entire code, such as third-party dependencies and eventually hidden vulnerabilities.

The first decision measure applied was to narrow down the number of available options by considering, according to Leitet [56], only the most popular toolsets: Vue.js [57], Ember.js [58], AngularJS [59], and React [60]. This decision was made following the assumption that major frameworks and libraries, in comparison to smaller players, are more likely to offer comprehensive and reliable functionalities in addition to well-written and accurate documentation. Moreover, the applicability of the web application for the professional software industry also contributed to this decision, as the use of the most in-demand tools aggregates value and interest to software systems.

Since the remaining four alternatives deliver similar functionalities, the subsequently applied selection criterion considered quality attributes which are of great importance for this project, particularly reliability and maintainability. While Ember.js and Vue.js are open source projects which rely on the contribution of individuals, AngularJS and React have been developed by mature companies, Google and Facebook, respectively. As such enterprises are more likely to offer maintenance and long-term support services for their products, only AngularJS and React continued as suitable candidates.

Finally, with two robust options remaining, the decisive selection criterion weighed their singular characteristics. An outstanding difference between React and AngularJS regards the approach utilised by each for solving the lack of congruence between static documents and dynamic applications. The former is, essentially, a collection of functions, also known as a library, that is called only when certain functionalities are required [60]. On the other hand, the latter is a structural framework that offers high-level of abstraction and great support for CRUD (create, read, update, and delete) applications at the cost of reduced low-level DOM (Document Object Model) [61] manipulation [62]. According to a performance experiment involving these two toolsets, React has proven to deliver better performance when it comes to rendering a large number of items [63]. Figure 4.11 shows how React efficiently updates DOM trees; it utilises a virtual React-DOM which is constantly compared against its own previous state, and only the necessary changes are applied to the actual DOM [64]. React was selected as the technology to be utilised for handling the creation of user interface components as its characteristics were perceived as more adequate for the implementation of this application.

4.2.2 Network Visualisation

Today, there are several JavaScript libraries compatible with web-based applications that support network (graph) visualisation. In some projects, choosing the appropriate tool(s) might not be as straightforward as thought, as it depends on system requirements and also on the interaction with other technologies already on place. A particular system constraint which plays a major role in such a decision-making process regards the size of the dataset to be handled by the application. The reason for such relevance is due to the technique utilised by most of the existing graph visualisation libraries; adding SVG elements in the browser's DOM tree. As Zimmer points out, this approach is suitable, in terms of

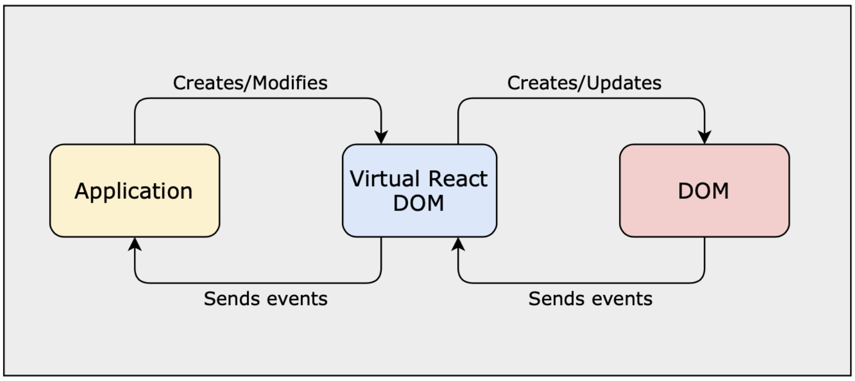


Figure 4.11: This image illustrates how React optimises rendering performance by means of a virtual DOM-tree.

performance, for graphs containing less than one thousand nodes [23]. Otherwise, if the purpose of the application is to handle large datasets, then the use of modern technologies is recommended, such as HTML5 Canvas- or WebGL-based technologies [65]. Taking into consideration the purpose and limitations of this study, as earlier emphasised, the use of SVG-rendering libraries is good enough.

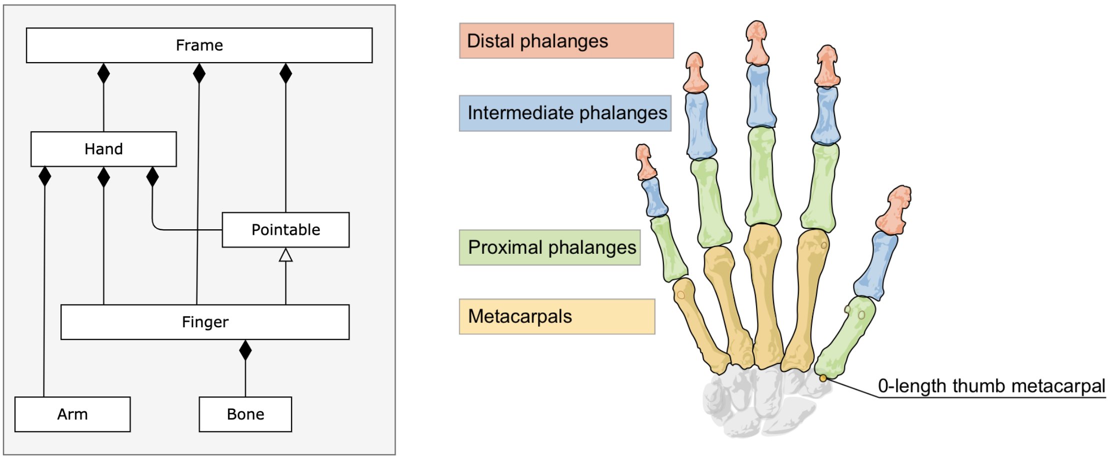
Following the recommendation given by the project supervisor, three JavaScript graph visualisation libraries were initially considered: Vis.js, Sigma.js, and D3.js. However, as the implementation phase unfolded, two other variants of such libraries that offer easier technology integration with React were identified: React-vis.js and React-sigma.js. To gain knowledge and understand the differences among the considered libraries, especially regarding functionality and performance, the trial and error problem-solving approach was sufficient to experiment with each of them.

The perceived outcome obtained from testing the different technologies revealed that the off-the-shelf React-wrapped libraries do not perform as the standard versions and/or impose undesirable limitations both to supported functionalities and to implementation design decisions. Therefore, the available alternatives were narrowed down to the standard variants of the libraries but recognising the drawback of having to adapt them to the React architecture. By further testing the remaining options, the conclusion was that D3.js was, overall, the most beneficial library for this project when considering aspects such as compatibility with React, documentation, learning curve, flexibility and scalability.

4.2.3 Leap Motion

LeapJS (or Leap.js) is the standard library utilised for the integration of the Leap Motion software with JavaScript applications. Therefore, there was no need for comparing and selecting from different alternatives as in the previously described cases. This project benefits the most from the library in terms of the many functionalities and mechanisms that it provides that make it simpler to work with the controller device, which optimises the implementation phase. For an application to connect and be able to receive motion tracking data captured by the hardware of a Leap Motion Controller device, LeapJS offers a Controller object that manages such communication. The connection is established through a local WebSocket server provided by the Leap Motion JavaScript SDK, which executes as a daemon or service, depending on the machine's operating system [66]. The communication follows a determined protocol, currently version v6.json, that specifies the format of JSON data transferred [67].

The tracking data is delivered from the controller to the application as a sequence of frames; each represents the temporary state of the system (recognised hands and fingers) at a unique instant in time. The LeapJS library implements a Frame object that contains all information about the elements identified in a given frame, including but not limited to the number of hands and fingers and their positions [12]. The library also supports functions that can be called by the application in order to obtain such frames, with the possibility of receiving them automatically upon recognition of motion by the hardware, or at specified time intervals [68]. Moreover, other classes also provided by the library, such as Hand, Finger, Bone, further support the obtainment of precise information about recognised human-body elements that are essential for the implementation of rich gestures. Figure 4.12a illustrates the classes supported by LeapJS that enable accessing and working with the different tracked entities and their relationships through standard UML notation. Figure 4.12b illustrates the finger bones that can be accessed through the Bone object, and their anatomical names.



(a) Classes and relationships

(b) Anatomical structure of fingers

Figure 4.12: The message transmitted by both images is that LeapJS provides the infrastructure required for the implementation of complex and rich gestures. While the diagram presented in (a) was designed according to the information provided in the official Leap Motion developer guide [11], the hand structure image in (b) was retrieved from the API overview source [12].

4.3 Implementation Overview

Given the identified requirements to be satisfied and the selected choices of technologies (frameworks and libraries) as the pillars of this application, the architectural structures of the system as a whole determines the achieved qualities of the system. To develop a web application and tackle the second technical requirement (TR 2), the architectural pattern selected was the client-server pattern. Taking into consideration the purpose, scope and limitations of this study, already in the early stages of development it was possible to conclude that there was no need for implementing business logic on an external server; instead, the server only supplies the index.html file and the implementation of all business logic is on the client. Such a design decision, when analysing the project in a long timeline perspective, could lead to performance issues as it is not suitable for handling large data sets. However, as the purpose of the system is to assist as an infrastructure for experiments, scalability was not regarded as a concern in consensus with the project supervisor. Therefore, the design decision of placing the business logic and processing all data on the client-side of the application is good enough and fit for the system.

Modifiability is an important quality attribute for this system since the commencement of the project, as it was anticipated that changes would be necessary throughout the implementation and experimenting phases. A relevant architectural decision was to build the application as a composition of blocks (or components) that communicate with each other according to a structured and quite restrictive guideline. This design enables separation of concerns and near-complete independence between modules in the system. In simple terms, the system implements a variant of the layered pattern, where a connector handles communication with the user interface component following a publish-subscribe approach. The modified layered pattern enables the system to be developed incrementally. Figure 4.13 shows the architectural structure of the client-side of the application.

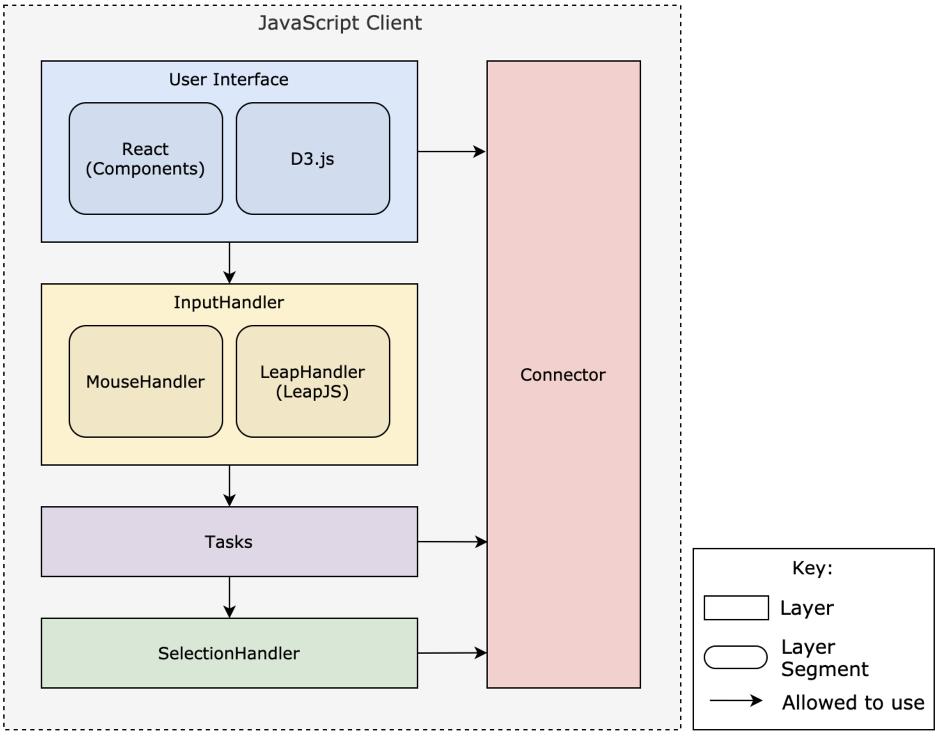


Figure 4.13: Architecture of the JavaScript client in a not-so-deep decomposition level. It is possible to observe how the system was designed according to a variant of the layered pattern.

Each layer in this architectural design contributes to the entire functionality of the system with a cohesive set of services. The User Interface module generates all the visuals for the user, including graphs, nodes, edges, texts and buttons which the user can interact with. This layer is allowed to subscribe to the Connector block, which notifies updates on the application data's state for changes to be reflected in the graphical representation. It is also allowed to interact with the InputHandler module, which receives inputs in the form of mouse events or hand gestures. In the InputHandler layer, each input is transformed and prepared into information that can be processed by the Tasks module, where the logic for all tasks, such as panning, zooming, and selecting a node, is located. The Tasks layer is allowed to communicate with the SelectionHandler layer, which stores the most recent and previous selection states. Both the Tasks and SelectionHandler layers communicate with the Connector module to send the latest publish information to be consumed by the User Interface.

The information flow that describes, from a coarse-grain perspective, the process of recognition and execution of actions performed through hand gestures is represented in Figure 4.14. The flow starts with a user triggering the Leap Motion controller by performing a hand gesture (user input). The controller hardware captures the movement and the Leap Motion SDK transforms the user skeleton information into understandable data according to the API specifications. The client application accesses and consumes the traking data. The InputHandler module receives this data and starts the gesture recognition process by accessing a storage of designed hand gestures. The module compares the received data against the stored data to find the correct task corresponding to the required user action. If there is a match between data, the module transfer data to the Task module that executes the task and updates both the SelectionHandler and the Connector layers with the outcome of the execution. Ultimately, the Connector notifies the User Interface of the changes so that it can be updated.

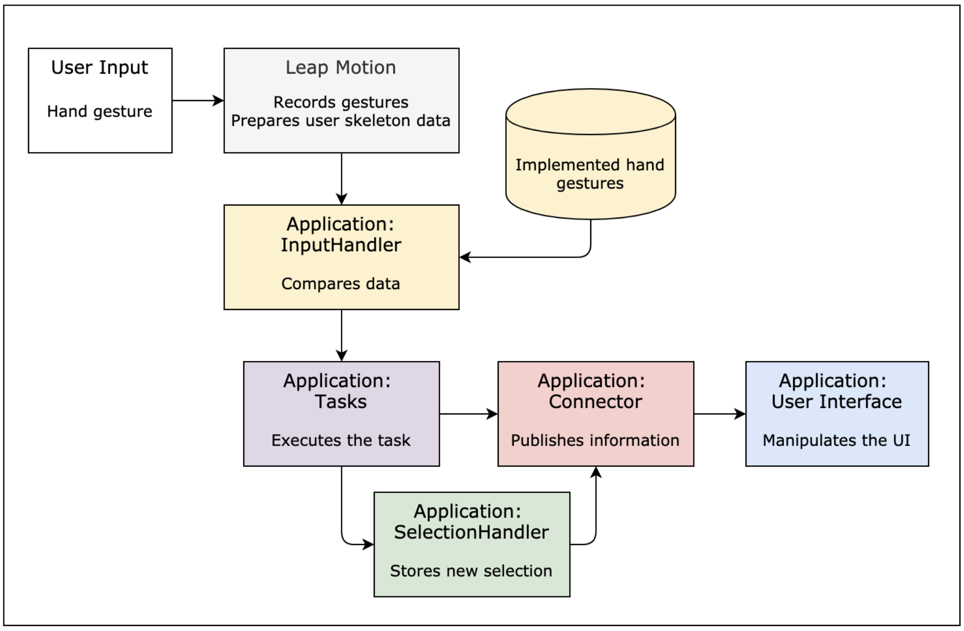


Figure 4.14: This diagram represents the end-to-end flow of information that takes place in the system when a user succesfully interacts with the application through hands gestures.

Besides the main application, it is also important to mention a recorder tool that was valuable for designing and implementing the hand gestures that populate the application storage, also identified in Figure 4.14. This tool was developed as a web-based application that receives data from a Leap Motion controller, transforms it according to a format compatible with the main application, and outputs the result of such transformation. The tool is simple; it does not employ any formal architectural structure, as its purpose is merely to be an infrastructure that optimises the implementation process. Therefore, no quality attribute, such as code quality or maintainability, was taken into consideration during its development.

4.4 Software Testing

Given the complexity, for the authors at least, of working with the leap motion technology, the necessity to manually test each gesture became more than evident. Manual testing, as the name itself implies, means that the people responsible for verifying the software test its functionalities in a non-automated approach (e.g., not using testing frameworks) by manually interacting with the software. In combination with this given approach of testing, exploratory testing was also conducted. In the latter, the people responsible for testing explore the software and its functionalities freely without constraints. By combining both testing techniques, each implemented task was verified in regards to the desired functionality and associated gesture to increase confidence that there was no faulty piece of code after each implementation phase.

The testing process was conducted iteratively, where each newly implemented task expressed a new iteration. Each iteration started with the manual testing of the hand gesture linked to that particular task. This first phase was carried out in an isolated environment to prevent conflicts with the already deployed source code. Then, the iteration proceeded with the manual testing of the functions assigned to that task. Finally, both the hand gesture and the functionalities were verified together. Once the manual testing phase was completed, the iteration advanced to the exploratory testing session. The implemented task—gesture and functionalities—was integrated and tested with other tasks. For each exploratory testing session, a graph data structure was randomly generated and examined with the aid of all tasks implemented so far. This process was repeated until no faults or errors were noticed. After the completion of each testing iteration, the authors progressed to the implementation of a new task.

5 Interfaces

This chapter introduces the network analysis tasks implemented during this study and their corresponding execution in both the gesture and mouse-based interfaces.

5.1 Gesture Interface

This section contemplates the project's second objective introduced in Section 1.4. As the result of literature reviewing studies that share a common background in HCI and related projects in the field of network analysis, the set of hand gestures developed for the execution of navigation and multivariate network analysis tasks are here reported. The design of such gestures took the aspects of comfort, popularity, intuitiveness, and recognition into consideration. The remaining portion of this section identifies the relationship between the implemented tasks and their corresponding hand gestures. For each gesture, textual description and graphical representation are provided.

5.1.1 Panning

The ability to panning across a network is accomplished through a hand gesture designed after the panning gesture implemented in the related work [10] and observed in the video [43]. The gesture was modelled to become, according to the authors' understanding, more intuitive for two-dimensionally displayed networks. Users activate the panning mode by opening their right hand above the Leap Motion Controller device with its palm pointing towards the computer screen. By maintaining such hand posture and moving it in different directions, as represented in Figure 5.15, users navigate a network in the corresponding direction to the executed motion.

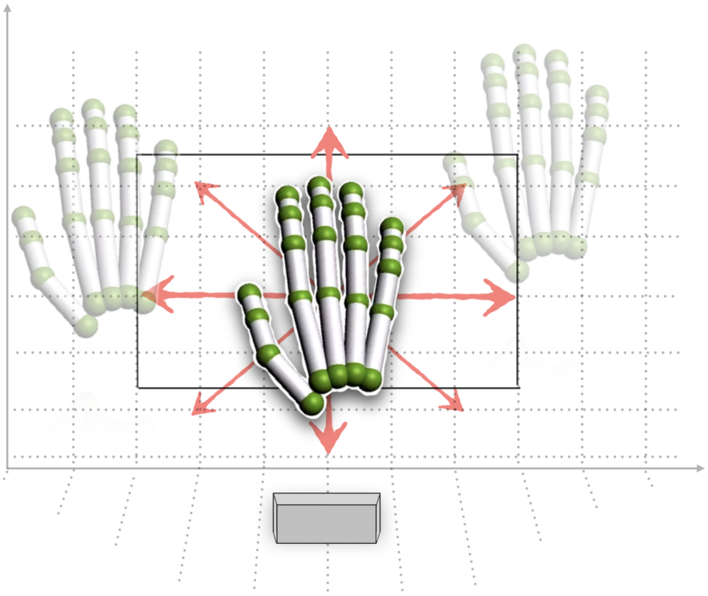
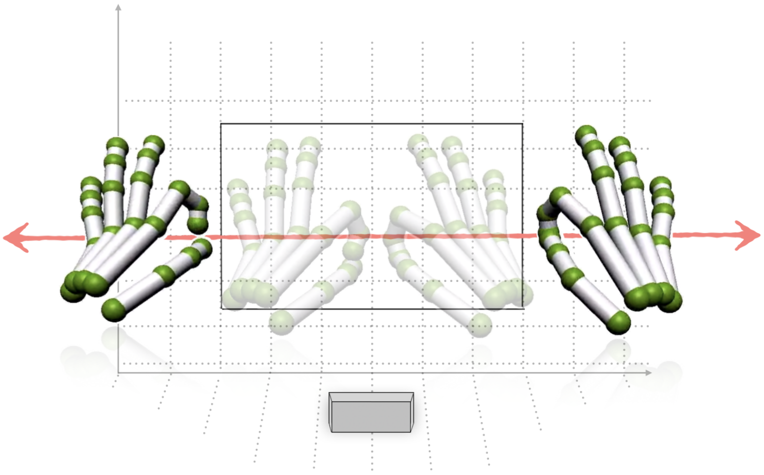


Figure 5.15: This image shows, from first-person perspective, the hand gesture associated with the panning task. The grey box symbolises the Leap Motion Controller device.

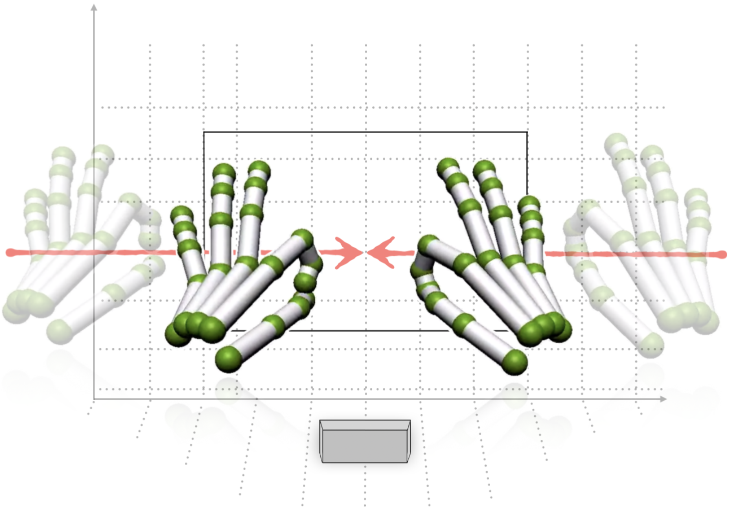
5.1.2 Zooming

The task of zooming an area of a network in and out is accomplished through two hand gestures inspired by the scaling gestures introduced in [7], such as Scale X-axis (1), Scale

Y-axis (1), and Scale Uniform (2) observed in Figure 1.5. Users activate the zooming mode by putting, above the Leap Motion Controller device, both their left and right hands in a pinch posture: thumb and index fingertips touching each other and the remaining fingers (fingers 3, 4, and 5) in extended position. It is recommended, for better recognition and comfort, that users produce such posture with their palms pointing towards an angle of $45°$ between the controller device and the computer screen. By maintaining the hands in such posture and moving them away from each other, users perform the zooming-in task, as represented in Figure 5.16a. By executing the exact opposite movement, moving the hands close to each other, users perform the zooming-out task, as represented in Figure 5.16b.



(a) This image demonstrates the gesture associated with the zooming-in task.

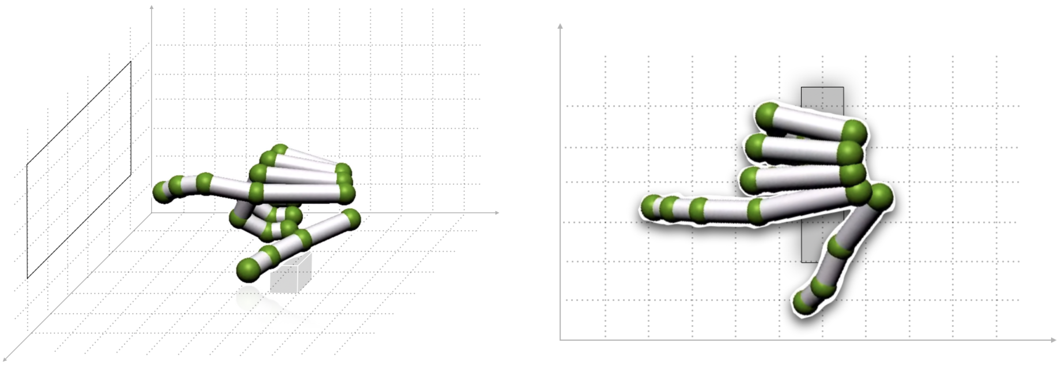


(b) This image demonstrates the gesture associated with the zooming-out task.

Figure 5.16: This figure shows, from first-person perspective, the hand gestures associated with the zooming task. The grey box denotes the Leap Motion Controller device.

5.1.3 Simple and Continuous Selection

The ability to select nodes and edges elements of a multivariate network, which can be achieved both by a repeated or a continuous motion, is accomplished through a gesture inspired by the click selection gesture, also known as index point gesture, presented in the research work of Lin *et al.* [69], where it was evaluated as the best gesture for selection of small two-dimensional objects performance-wise. The implemented gesture modifies the referenced gesture by rotating the right-hand wrist in 90° to the left (e.g., right hand on horizontal axis instead of on vertical axis) to, according to the authors' understanding, enhance comfort and ease recognition. Users activate the selection mode by making the following posture above the Leap Motion Controller: right-hand thumb and index finger extended (forming an L or V shape) and the remaining fingers (fingers 3, 4, and 5) folded, as displayed in Figure 5.17.

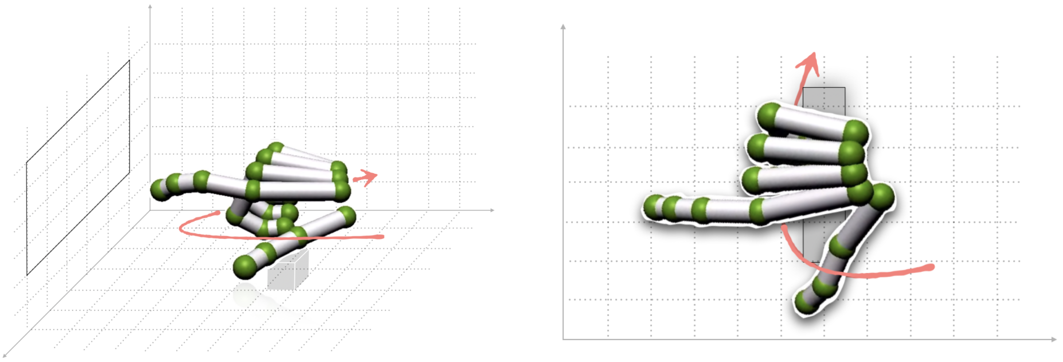


(a) Side perspective

(b) Above perspective

Figure 5.17: This figure illustrates, from both side and above perspectives, the initial hand posture associated with the selection task. The grey box denotes the Leap Motion Controller device.

Maintaining such a posture, users are enabled to move their right hand until the Leap Motion cursor reaches a target (network element) of interest. Then, to trigger the selection task, users fold their thumb inwards their palm until it touches the side of their curled middle finger (finger 3), motion represented in Figure 5.18.

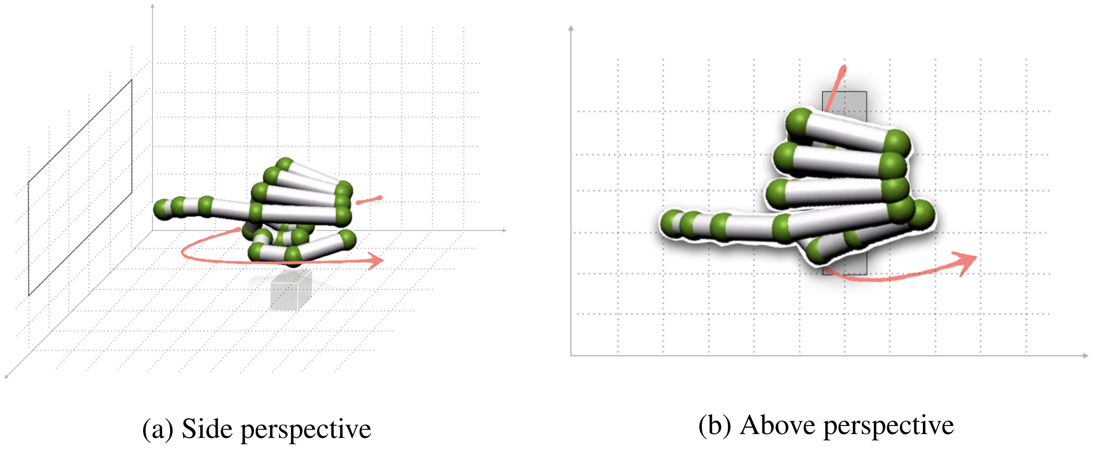


(a) Side perspective

(b) Above perspective

Figure 5.18: This figure illustrates, from both side and above perspectives, the thumb folding motion required to trigger the selection task from the initial hand position. The grey box symbolises the Leap Motion Controller device.

Users, with their thumb touching on the side of the middle finger and their index finger pointing at the desired target, have the options to either keep the thumb in the described position (continuous selection mode) or return the thumb to its initial extended position (simple selection mode), motion illustrated in Figure 5.19. In the continuous selection mode, users, by maintaining such posture and moving their right hand in different directions, select all elements with which the Leap Motion cursor comes in contact. To leave the continuous selection mode, users return their thumb to its initial extended position. On the other hand, in the simple selection motion, users can select various network elements by repeating such thumb tapping gesture targeting the desired elements.


(a) Side perspective

(b) Above perspective

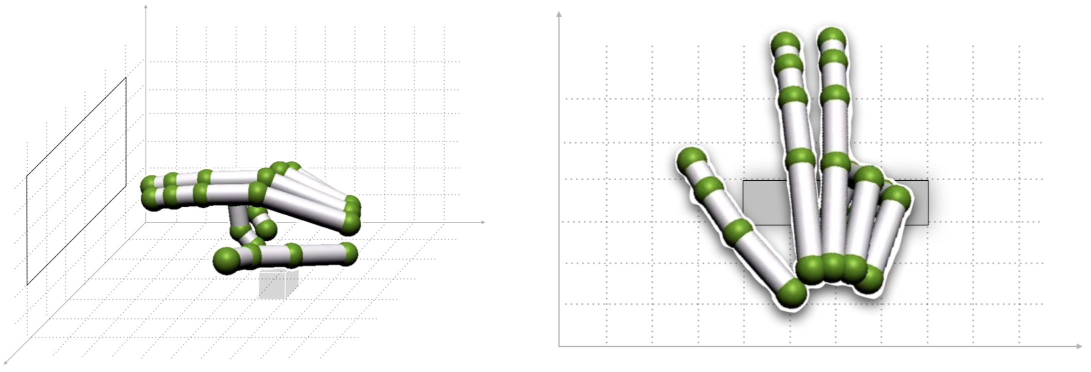
Figure 5.19: This figure illustrates, from both side and above perspectives, the thumb returning motion required to quit both the simple and continuous selection tasks. The grey box denotes the Leap Motion Controller device.

5.1.4 Area Selection

The ability to select all nodes within an outlined area of the network, a task referred to as area selection, is accomplished through a hand gesture similar to the selection gesture just described. The decision of creating a related gesture instead of designing a dissimilar one took into account (1) the performance of the gesture utilised as a reference for the selection task, and (2) the learning curve for users. Users activate the area selection mode by setting, above the Leap Motion Controller device, their right-hand thumb, index and middle fingers in an extended posture and their remaining fingers (fingers 4 and 5) folded, as presented in Figure 5.20.

Then, to begging drawing the desired area of selection, users fold their thumb inwards their palm, under their index and middle fingers, until it reaches their finger 4, according to the motion illustrated in Figure 5.21. After executing such motion and arriving at the posture shown in Figure 5.22, users are ready to start outlining the area of interest. By maintaining such hand posture, with the thumb touching the side of the finger 4, users should outline the area in which nodes within will be selected at the end of the gesture. This task is similar to the well-known lasso selection functionality in image editing software. The drawing motion should not be too fast and, ideally, should have juxtaposed start and endpoints. Otherwise, the application completes the area outlining with a straight line between these points, which often ends up producing an outcome different from the expected. Figure 5.23 represents such a gesture.

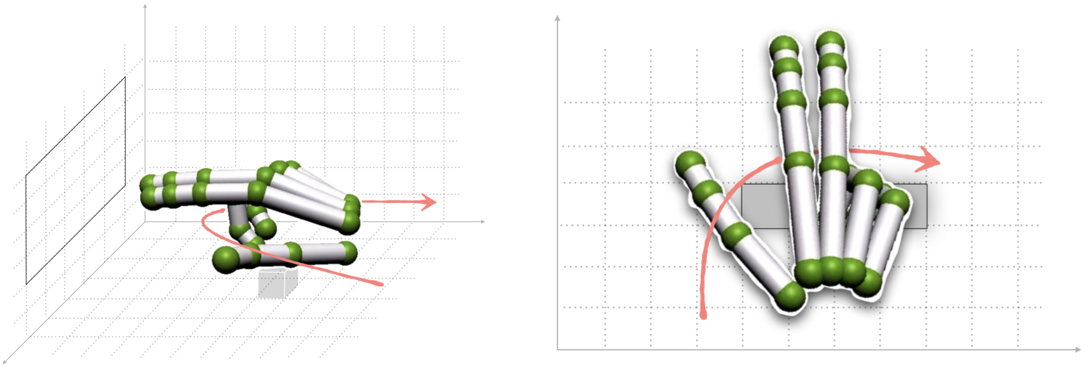
Finally, to conclude the drawing action and terminate the area selection task, users return their thumb to its initial extended position, as represented in Figure 5.24, arriving back at the initial posture of the gesture.



(a) Side perspective

(b) Above perspective

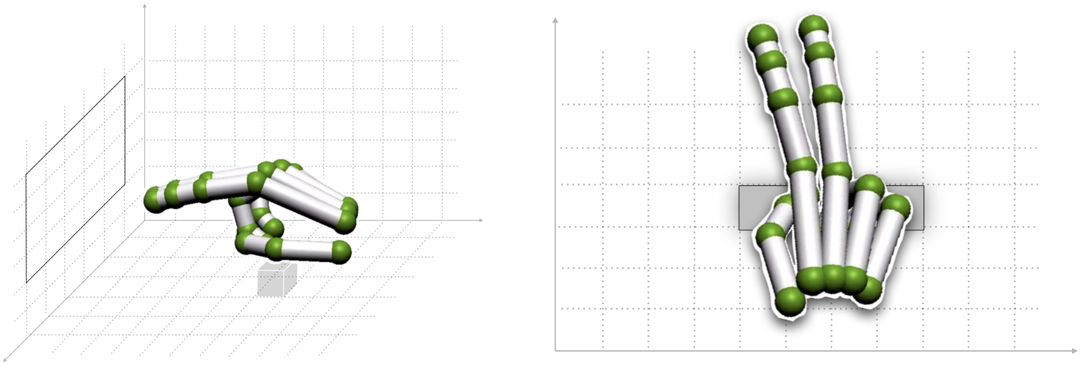
Figure 5.20: This figure illustrates, from both side and above perspectives, the initial hand posture associated with the area selection task. The grey box denotes the Leap Motion Controller device.



(a) Side perspective

(b) Above perspective

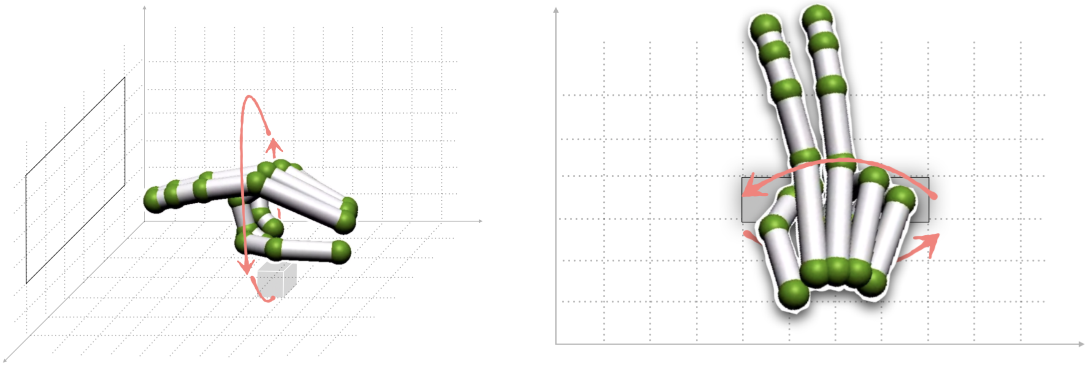
Figure 5.21: This figure illustrates, from both side and above perspectives, the thumb folding movement required to start the area drawing action. The grey box indicates the Leap Motion Controller device.



(a) Side perspective

(b) Above perspective

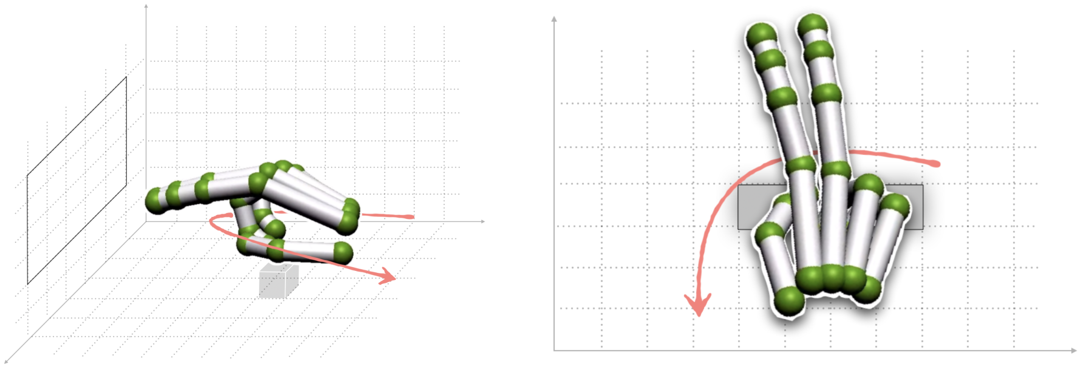
Figure 5.22: This figure illustrates, from both side and above perspectives, the posture which users arrive after the thumb folding motion is completed. The grey box symbolises the Leap Motion Controller device.



(a) Side perspective

(b) Above perspective

Figure 5.23: This figure represents, from both side and above perspectives, the hand motion which users perform to draw the desired selection area. It is important to notice that the motion can be executed both in a clockwise or anti-clockwise direction. The grey box denotes the Leap Motion Controller device.



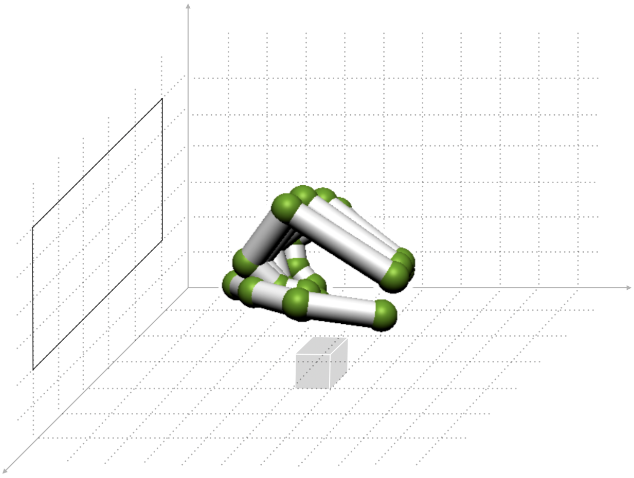
(a) Side perspective

(b) Above perspective

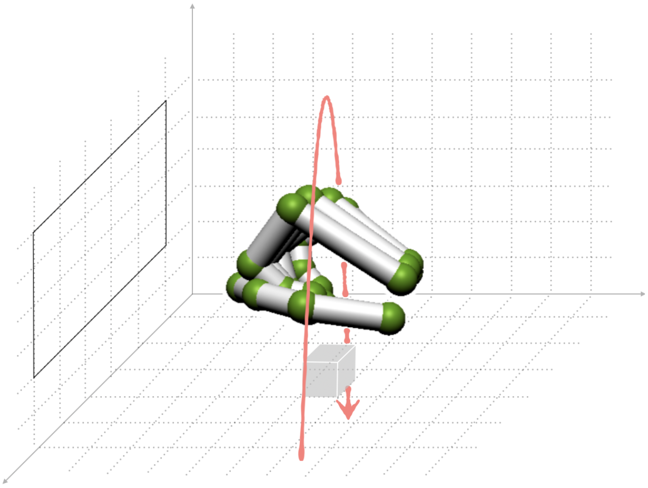
Figure 5.24: This figure represents, from both side and above perspectives, the thumb returning motion required to terminate the area selection task. The grey box indicates the Leap Motion Controller device.

5.1.5 Deselection

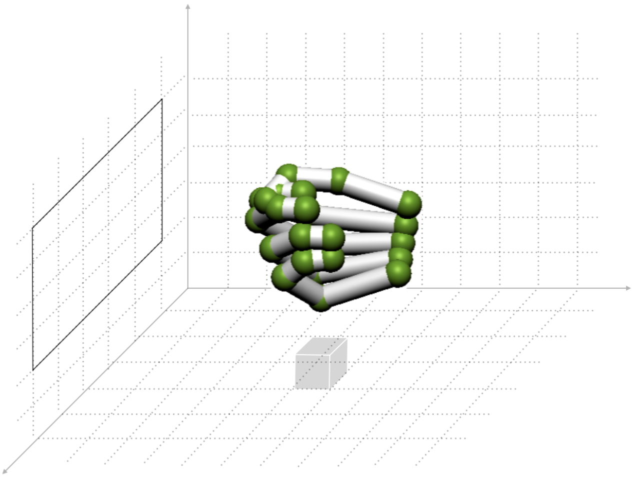
The ability to deselect all currently selected network elements, both nodes and edges, is accomplished through a hand gesture designed after two gestures introduced in [7], namely Delete (1) and Stop (2) as observed in Figure 1.5. Users activate the deselection by showing, above the Leap Motion Controller device, their right hand in a horizontal clenched fist posture: closed hand with fingernails pointing towards the tracking device and knuckles pointing towards the computer screen, as illustrated in Figure 5.25a. From such clenched fist posture, users proceed with the gesture by rotating their right wrist in an 90° to the right, motion demonstrated in Figure 5.25b. The entire gesture, from its beginning with setting the horizontal clenched fist posture to its completion with the rotation motion, should be a continuous, quick, firm and sharp movement, without the need of waiting on the initial fist pose. The conclusion of the gesture is achieved when users arrive at the vertical clenched fist posture, as illustrated in Figure 5.25c. The result of the deselection task is a network where nothing is selected nor highlighted.



(a) This image illustrates the hand posture at the beginning of the motion.



(b) This image demonstrates the rotation motion.

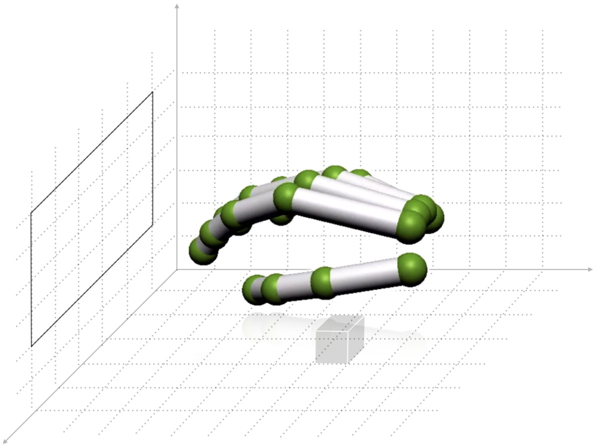


(c) This image illustrates the hand posture at the end of the motion.

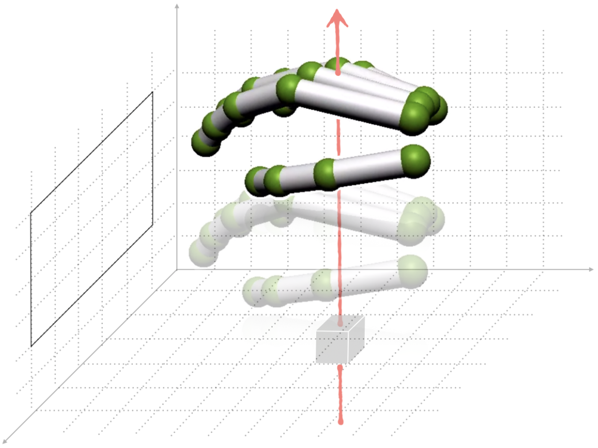
Figure 5.25: This figure illustrates, from side perspective, the hand gesture associated with deselection task. The grey box indicates the Leap Motion Controller device.

5.1.6 Filter Menu

The task of accessing the filter menu is accomplished through a hand gesture inspired by the VM Open gesture introduced in [7] observed in Figure 1.5. The implemented gesture modifies the referenced gesture to enhance comfort, according to the authors' judgment. Users start the open filter gesture by placing, above the Leap Motion Controller device, their right hand fully opened (fingers extended and palm pointing the controller device), as illustrated in Figure 5.26a. By maintaining such hand posture and lifting it in a vertical axis with the controller device, motion represented in Figure 5.26b, users complete the gesture and gain access to the filter menu.



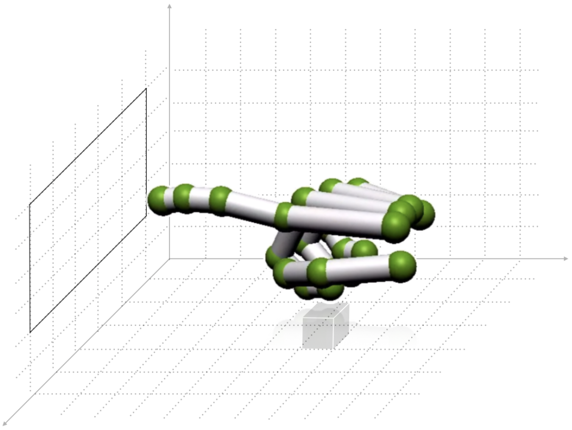
(a) This image illustrates the hand posture at the beginning of the motion.



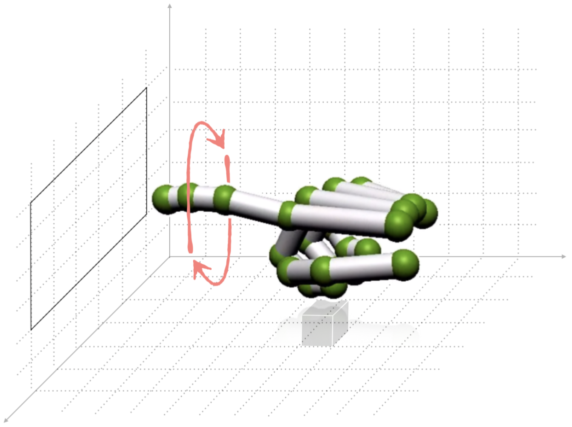
(b) This image demonstrates the lifting motion.

Figure 5.26: This figure illustrates, from side perspective, the hand gesture associated with the open filter task. The grey box indicates the Leap Motion Controller device.

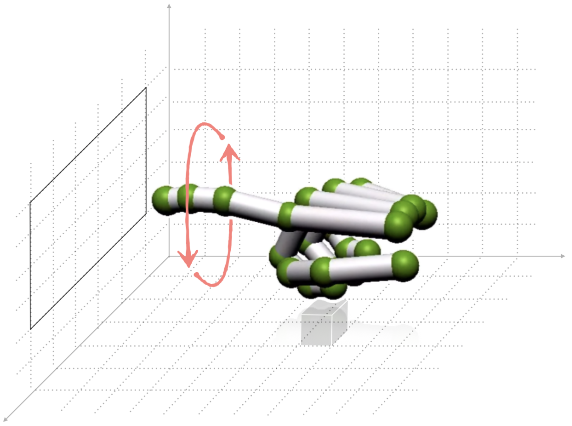
Once the filter menu is open, users interact with its different options according to the provided instructions. The task of defining a specific value for a condition, which can be either a string or an integer value, in the filter functionality is accomplished by a finger rotation gesture. Before executing such gesture, users must have (1) the Leap Motion cursor targeting the value spinner and (2) their right-hand index finger extended and the other fingers folded, as illustrated in Figure 5.27a. When (1) and (2) are met, users increase values by spinning the extended index finger to the right, as represented in Figure 5.27b, or decrease values by spinning to the left, as represented in Figure 5.27c.



(a) This image illustrates the hand posture at the beginning of the motion.



(b) This image demonstrates the spinning motion to the right.



(c) This image demonstrates the spinning motion to the left.

Figure 5.27: This figure illustrates, from side perspective, the hand gesture associated with the task of changing values. The grey box indicates the Leap Motion Controller device.

When a value is chosen for a given filter condition, users confirm the operation by performing the thumbs-up gesture. In this gesture, users maintain, above the Leap Motion Controller device, their right hand in a thumbs-up signal for a short period, posture illustrated in Figure 5.28.

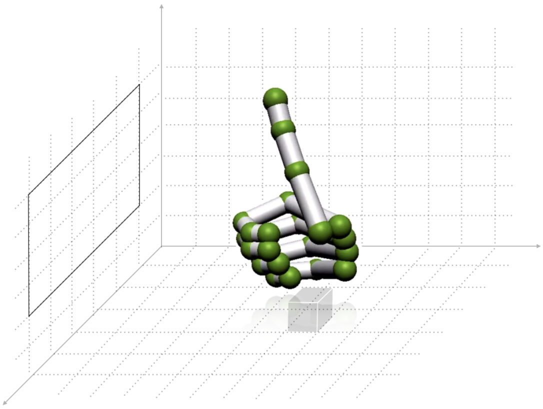
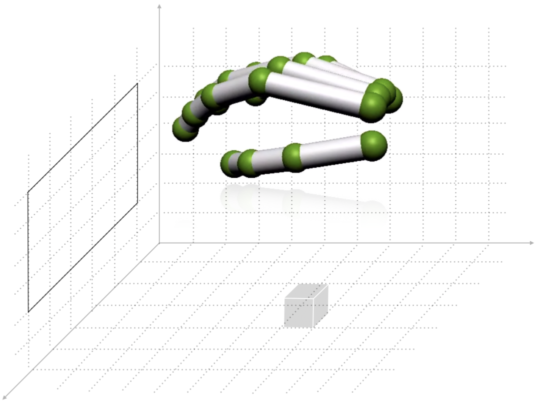
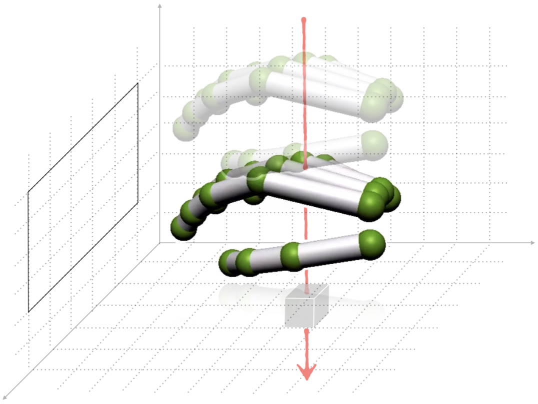


Figure 5.28: This image illustrates, from side perspective, the hand posture associated with the thumbs-up gesture. The grey box symbolises the Leap Motion Controller device.

Users start the close filter gesture by placing, far above the Leap Motion Controller device, their right hand fully opened (fingers extended and palm pointing the controller device), as illustrated in Figure 5.29a. By maintaining such hand posture and lowering it in a vertical axis with the controller device, as represented in Figure 5.29b, users complete the gesture motion and close the filter menu.



(a) This image illustrates the hand posture at the beginning of the motion.



(b) This image demonstrates the lowering motion.

Figure 5.29: This figure illustrates, from side perspective, the hand gesture associated with the close filter task. The grey box indicates the Leap Motion Controller device.

5.1.7 Relocate

The ability to relocate, as the task of revisiting previous selection events is referred to, is accomplished through a hand gesture created after the Previous gesture introduced in [7] and observed in Figure 1.5. Users trigger the relocate task by performing, above the Leap Motion Controller device, a left-to-right swipe motion with their right hand. The hand posture and motor motion associated with this gesture can be observed in Figure 5.30. Moreover, the swipe movement should be a continuous and quick motion.

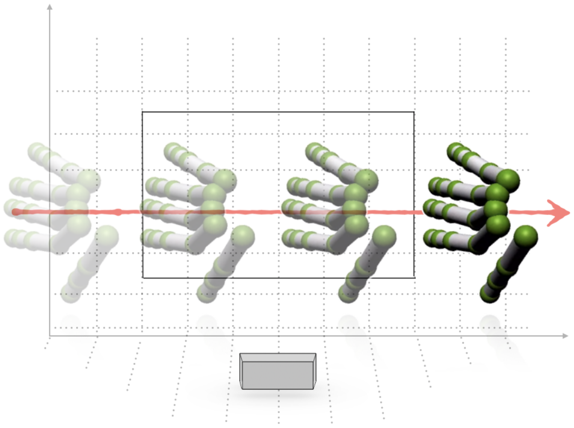


Figure 5.30: This image shows, from first-person perspective, the hand gesture associated with the relocate task. The grey box denotes the Leap Motion Controller device.

5.1.8 Find Shortest Path

The ability to find the shortest path between two selected nodes is accomplished through a hand gesture created after the Group gesture introduced by Huang *et al.* in [9] and observed in Figure 2.7. Users begin the gesture by placing, above the Leap Motion Controller device, both their left and hands opened, thumbs pointing upwards and fingers 2, 3, 4, and 5 pointing towards the computer screen, with a space between the hands. Then, users perform a clapping motion to complete the gesture, as shown in Figure 5.31.

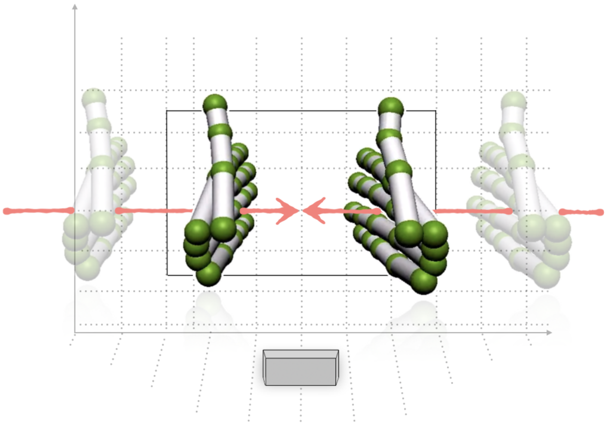
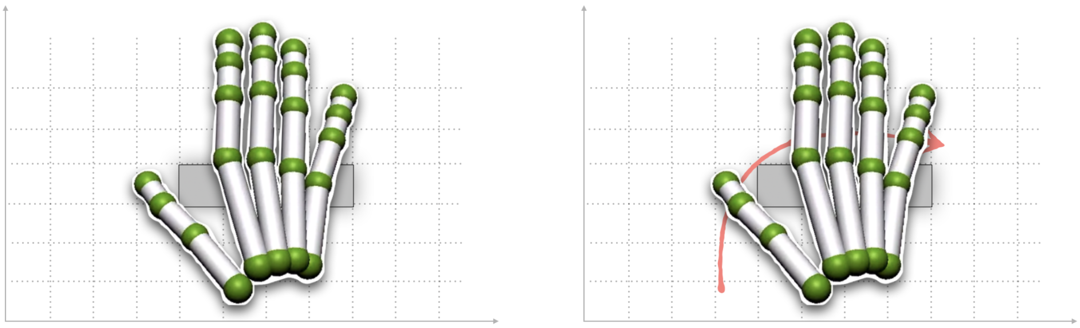


Figure 5.31: This image represents, from first-person perspective, the gesture associated with the find shortest path task. The grey box denotes the Leap Motion Controller device.

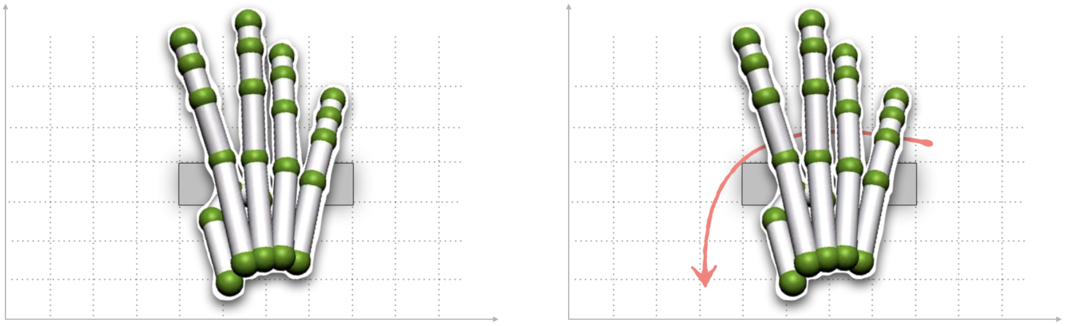
5.1.9 Find Adjacent Nodes

The ability to find all adjacent nodes of a single or group of selected nodes is accomplished through a hand gesture similar to both the previously disclosed selection gestures. The decision to further extend the mentioned hand gestures instead of creating a new one took into account the learning curve as well as the intuition aspect. Users commence the find adjacent nodes gesture by setting, above the Leap Motion Controller device, their right hand open with all fingers extended and palm facing downwards, as illustrated in Figure 5.32a. Then, to trigger the find adjacent node functionality, users fold their thumb underneath their palm (motion represented in Figure 5.32b) to reach the hand posture represented in Figure 5.32c. Finally, to conclude the gesture, users return their thumb to its initial extended position, motion represented in Figure 5.32d. The folding and releasing thumb movement should be quick and sharp, resembling a tapping motion.



(a) Initial hand posture

(b) Folding thumb motion



(c) Intermediate hand posture

(d) Releasing thumb motion

Figure 5.32: This figure represents, from above perspective, the hand gesture associated with the find adjacent nodes task. The grey box indicates the Leap Motion Controller device.

5.2 Mouse Interface

After describing the hand gestures associated with each of the implemented tasks, it is reasonable to also describe their corresponding mouse-based implementation. As this is a study built upon a comparison between both interfaces, the relevance of such information comes from the fact that lower results can be a consequence of either a true difference between the interaction styles or simply a more cumbersome choice of interaction in one side. In the mouse-based interface, most of the tasks are accomplished following a similar execution manner to keep the interface as straightforward and intuitive as possible.

For the tasks that are regularly exercised in various software and most likely familiar to the users (e.g., Selection, Zooming and Panning), standard execution approaches were implemented. It means the mouse execution resembles the manner in which these tasks are performed in the other applications. For example, the Selection task is achieved by left-clicking on the desired entity, the Panning task by the click and drag motion on the network background, and the Zooming task by scrolling the mouse scroll wheel. Nevertheless, no literature review was conducted to identify such interactions; the authors' experience and common sense were preponderant in determining them. On the other hand, for the remaining tasks which have no obvious corresponding mouse execution (e.g., Find Shortest Path, Relocate and Find Adjacent Nodes), a menu was developed, as illustrated in Figure 5.33. It is accessed by right-clicking on the network background, and the interaction continues with the selection of the desired task. The Find Adjacent Nodes task, for example, is accomplished by first selecting a node followed by accessing the menu and choosing the corresponding option. It is relevant to mention that no shortcut keys were implemented to avoid the use of a keyboard.

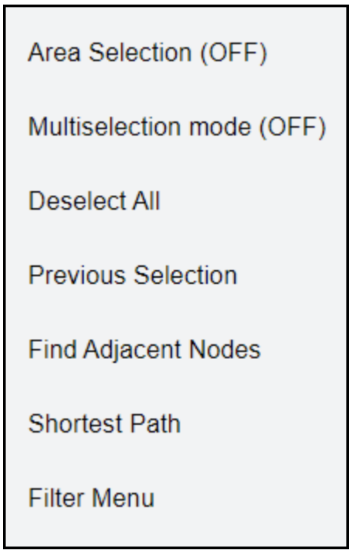


Figure 5.33: This image illustrates the implemented menu for the mouse-based interface. It enables users to perform tasks which have no commonly associated execution patterns.

6 Expert Review

This chapter describes the environment and the manner in which the expert review, mean of evaluation of the designed hand gestures and developed system, was conducted as well as reveals the feedback obtained at the completion of the evaluation process.

6.1 Description

The expert review activity was planned and executed following a series of precautions to create and deliver the most pleasant participation experience possible to the expert. The first measure towards achieving this goal considered, due to the previously described COVID-19 circumstances, the physical environment where the activity took place. The authors, searching for a solution to provide a safe environment for the expert, decided, in agreement with the project supervisor, to conduct such review activity in a virtual environment, where everyone could join from their respective homes. The chosen approach introduced further challenges which also had to be overcome, such as the establishment of a continuous communication channel between the authors—the investigators—and the expert as well as the preparation of the experiment to operate on a different computer from where the application was developed, which includes variations of Leap Motion Controller devices and software development kit versions.

The project supervisor established the initial contact between both parties—the expert and the investigators—and, from that moment, the communication proceeded through email. The second measure targeting the expert experience regarded information awareness. The expert was contacted and provided with comprehensive details about both the project schedule and the practical aspects of the review procedure, including his role as the expert, the tasks expected to be fulfilled, and the subsequent steps. Both parties agreed on a date and time for the online meeting, which was conducted using Zoom [70] as the platform for video conferencing, according to the expert's suggestion. On the day preceding the experiment, the expert was given access to the project report, where information and description of the implemented hand gestures could be found, and also to the questionnaire to be answered throughout the experiment, see Appendix A.

The third measure of the series of precautions took place during the meeting. At the beginning of the experiment, the expert was presented with a demonstration where the investigators went through the available tasks, in the same order as written in the report, and executed the hand gestures associated with such tasks. This demonstration was performed in play mode to avoid revealing, before the commencement of the actual review activity, the network topology (data set) associated with the questionnaire. This data set was designed to suit, with some degree of complexity, the tasks in the questionnaire, creating non-trivial scenarios for the expert when solving the questions, see Figure 6.34. It was considered the total number of nodes, connected and disconnected groups of nodes, the maximum and the minimum number of edges between nodes, and the attributes. Then, the expert was given the opportunity to ask related questions and, finally, was provided with the Uniform Resource Locator (URL) of the application for the evaluation. Before the investigators left the Zoom meeting, the expert was informed that the investigators would be online and available through email to answer follow-up questions during the remaining of the experiment. Lastly, the expert was instructed to, before going through the questionnaire, practice the gestures and mouse-based interactions in the play mode. The reason for this instruction was to reduce possible validity problems of biased results associated with the learning curve of first executing a task in one interface and then on the another.

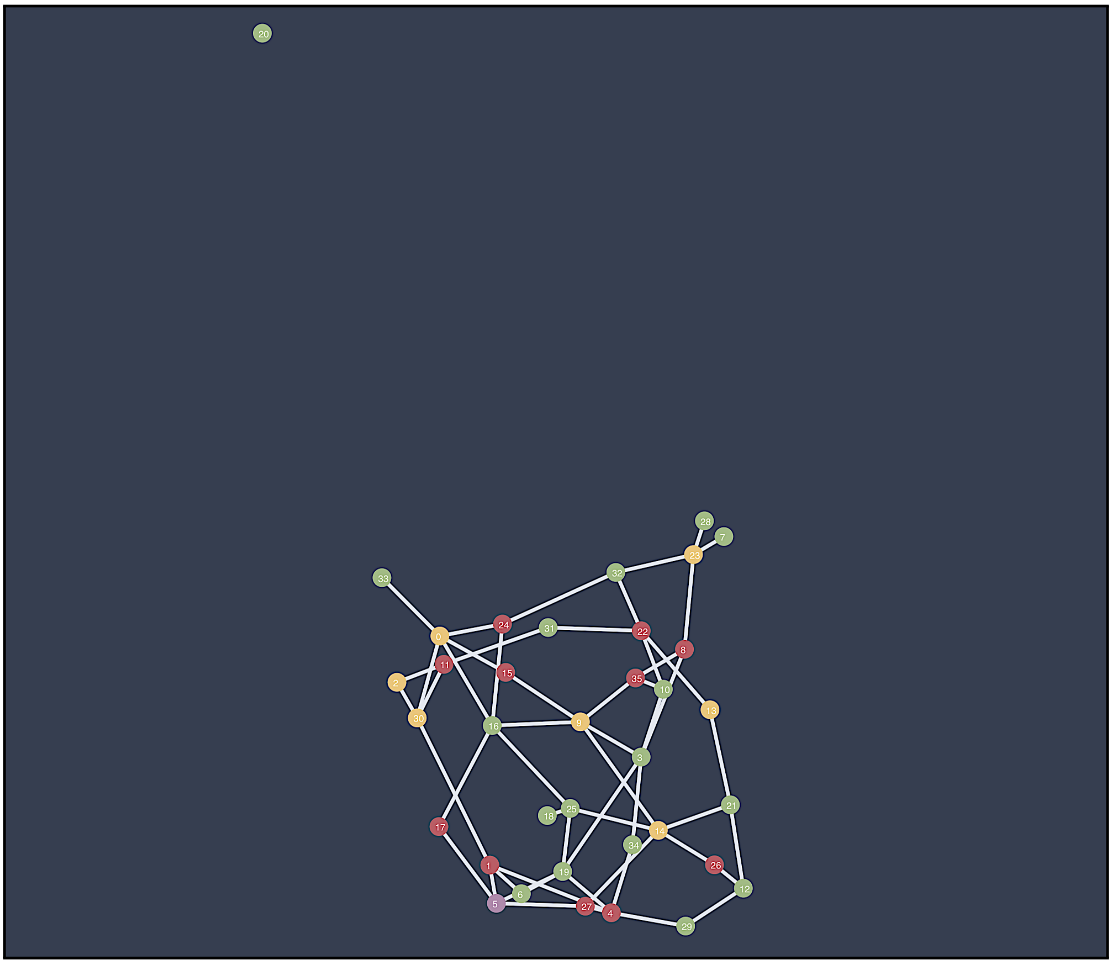


Figure 6.34: This image exhibits the network topology to which the questionnaire was applied. The data set represents a social media network and contains 35 nodes. It contains a unique disconnected node. The number of edges for a node ranges from 1 to 5, where the maximum corresponds to approximately 14,3% of the total number of vertices in the network. Nodes are colour-encoded according to the value of their age attribute.

Besides answering the questionnaire, the expert was also sought on the possibility of providing video files containing a recording of the computer screen as well as of the hand gestures during the experiment, which was consented. The first reason for requesting such recordings was to ensure that the expert followed and accomplished the tasks correctly. The second reason was to extract information related to accuracy and ease of execution which perhaps could not be expressed with words in the questionnaire. The results obtained from the questionnaire and video files are located in Subsection 6.2.

The expert who contributed to this research is a person with significant background in the fields of human-computer interaction and media technology, including more than seven years of experience when taking into account both study and work time. To the moment of the experiment, the expert worked at the Department of Computer Science and Media Technology at Linnaeus University, with responsibilities for idea conceptualization, software development, planning and conducting user interaction studies with human participants, and evaluation. The focus of recent research conducted by the expert in the field of human-computer interaction targeted mostly immersive analytics, virtual reality, three-dimensional user interfaces, and computer-supported collaborative work.

The computer environment utilised by the expert to access and evaluate the system had the following specifications:

- Operating System: Windows 10 Home
- Display: DELL U2715H, 2560x1440
- Browser: Chrome version 81.0.4044.122 (Official Build) (64-bit)
- Leap Motion Software version: 2.3.1+31549
- Leap Motion Firmware revision: 1.7.0
- Optical Mouse: Logitech MX 518

6.2 Results

This subsection reports the feedback provided by the expert. While the first part presents the results of the numerical evaluation of each of the tasks, the second part discloses the specific comments that support such attributed values. The purpose of collecting quantitative data, even when considering that only one person evaluates the system, is to quantify the experience and opinions of the expert. In this way, the authors' obtain a more tangible object for the comparison purpose than words (e.g., the expert may express issues with a particular gesture in the comments section but still enjoy the overall user experience). Finally, the last part contains the overall impression and improvement suggestions given by the expert. The results here reported are presented in raw format; they are objective and direct.

6.2.1 Numerical Evaluation

The expert rated the usability experienced throughout the execution of the tasks, for both mouse device and hand gesture interactions, by numerical representations ranging from 1 (horrible) to 5 (excellent). Table 6.2 displays how the expert attributed values to each of the tasks, while Figure 6.35 graphically represents the obtained result.

Task	Rating	
	Mouse Device	Hand Gesture
Panning, Zooming, Hoovering	4	4
Simple Selection	4	3
Continuous Selection	4	4
Deselection	3	3
Area Selection	4	3
Find Adjacent Nodes	4	3
Open Filter	4	4
Select Value	4	3
Confirm Value	4	4
Close Filter	4	4
Find Shortest Path	3	4
Relocate	2	4

Table 6.2: Numerical evaluation ranging from 1 (horrible) to 5 (excellent) for each of the tasks present in the questionnaire. The ratings reflect the expert's usability experience.

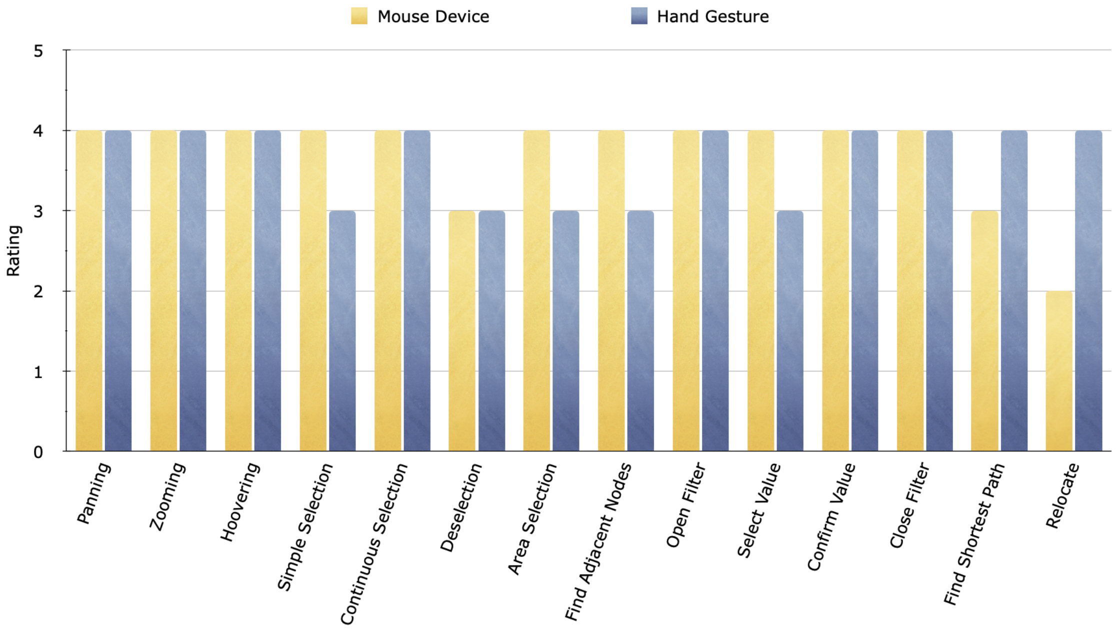


Figure 6.35: Graphical representation of data presented in Table 6.2.

6.2.2 Expert Comments

The expert, besides providing a numerical evaluation for each interaction style in each task, also presented comments supporting such evaluation decisions. These comments are arranged and grouped in accordance with the tasks they refer to. In each of the subsequent items, the symbol (MD) indicates the beginning of the comments describing the usability experience with the mouse device interaction interface, while (HG) indicates the same but for hand gesture.

- **Panning, Zooming and Hoovering:**
  (MD) “Straight-forward mouse wheel and dragging.”
  (HG) “Implemented gesture interaction for panning, zooming, and hovering worked fine and as expected. I particularly liked panning and zooming, as it seemed fairly reliable and responsive. Depending on the zoom level, it may be a bit more tricky to hover over a node (zoomed out -> nodes appears smaller), as (1) one rarely holds a finger completely still, and (2) there is always a bit of “gitter” in the detection with these kinds of interfaces (which is to be expected).”
- **Simple Selection:**
  (MD) “Straight-forward.”
  (HG) “Hovering over the node and selecting it with the quick thumb movement felt fine. Maybe one need to perform the gesture once or twice until the desired selection is applied, but since the gesture itself is so “easy”, I don’t see a problem.”
- **Continuous Selection:**
  (MD) “Straight-forward.”
  (HG) “For edge selection, I ended up using the implemented functionality of continuous selection (maintaining the thumb posture so that everything that is hovered, is added to the selection), as it was easier to select the edge like that than the “first hover, then select” approach as used before (since realistically speaking one ends up moving the finger a bit during the thumb gesture, ultimately moving the pointer so that nothing

is hovered). Consequently, I think it was great that this interaction option was provided, enhancing the gesture interface.”

- **Deselection:**
  (MD) “Right-click in empty space or context menu, select “Deselect All” to “On”.”
  (HG) “Fist-rotation interaction required a few tries, but worked fine.”
- **Area Selection:**
  (MD) “Right-click in empty space for context menu, select “Area Selection” to “On”. Then click the nodes.”
  (HG) “Generally, the interaction worked quite well. However, in my case at first it felt that there were some minor sensor detection issues, which lead to an abortion of the gesture midway (while I wasn’t done). Again, this is a common bottleneck with these sorts of interfaces, which unfortunately may lead to some (minor) frustration with the user (if it happens too often). Consequently, this is something to look out for in the future (how to make it more reliable on the software side). Independent of that, I liked the ease of use / easy to learn combination of using two extended fingers and the thumb-movement for confirmation.”
- **Find Adjacent Nodes:**
  (MD) “Right-click on one of the selected nodes from task 5, and select “Find Adjacent Nodes”.”
  (HG) “The gesture itself worked fine. However, I would recommend to provide the user with some (visual) feedback, e.g., as a timer, stating that the gesture is detected and ongoing, as the interval for triggering the interaction felt (subjectively) a bit long”.”
- **Filter:**
  (MD) “I think here it would be great if the filter could be applied “automatically” to all nodes, when “none” are selected.”
  (HG) “The interaction worked fine. It felt that one had to rotate the index finger a fair amount to come to the “Sweden” option, but that is fine in my opinion within this application. However, one need to be careful to not require repeating gestures that might “strain” the user’s hand.”
- **Find Shortest Path:**
  (MD) “Right-click for context menu and selected “Shorted Path”.”
  (HG) “The clapping was fine. I wasn’t sure if both hands were detected at the start (see general comment in regards to the application’s status and user feedback below).”
- **Relocate:**
  (MD) “Solved using a couple of iterations of “Right-click in empty space for context menu, and select Previous Selection”.”
  (HG) “The swiping gesture with the right hand worked perfectly fine.”

6.2.3 Overall Impression

The final section of the questionnaire enabled the expert to express the overall impression of the application in general, emphasize comments about the interaction interfaces, provide positive feedback and report issue, and provide improvement suggestions. The answer obtained was divided into sections, as follows.

- **Mouse device interaction:**
  “Generally, pretty straight-forward and as one would expect.”

- **Hand gestures interaction:**
  “Filter menu: It would be great to show the index options (i.e., the number of fingers required to select an option in the filter menu) more prominently (not at the end of the option’s text description).
  Note: There was the unexpected behaviour were the filter menu could not be applied, as the loading for the confirmation gesture/posture was not loading — the developers provided a fix before the task completion.
  Leap Motion cursor: For me, my normal pointing hand posture features the thumb in the folded (non-extended) position. Therefore, having it extended most of the time while placing the pointer felt a bit unnatural for me. From own experiences and observations working with gesture interfaces, I can also relate that different users have different “base” positions in regards to holding or folding their hand posture. Maybe one can consider some kind of initial preferences calibration of the user (future work). Nevertheless, the interaction worked fine.
  Provide visual feedback for which and how many hands are detected, in a sense that there is only one pointer, and that one only moves with the respective detected gestures. Consequently, it would be good for the user to know additional information about the state of the application, for instance, to know if the hands are detected in the Leap Motion controller’s field of view. Maybe a small status display in the corner of the application could help with that.
  Generally, the provided gestures were easy to learn and perform. Some inaccuracies in the sensors detection is normal. I also sometimes felt that there was a quite delay between my (the user’s) interaction and the detection translation of the action in the application (this is very subjectively speaking; keep in mind that I am working with virtual reality applications where the responsiveness has to be spot on, which might make me more sensitive to these sort of things). All the tasks could be completed. Overall, I enjoyed completing the tasks and playing around with the application using the gesture interface.”

7 Discussion

This chapter analyses the expert feedback introduced in the previous chapter and discusses the conclusions that can be drawn from it. Before starting, however, it is important to mention that some of the findings here reported cannot be entirely backed up the expert review data. In such cases, when there is no concrete evidence for a statement, the authors explicitly inform the subjectiveness of such a conclusion. The chapter begins with high-level conclusions and then analyses the most interesting cases individually.

The first conclusion drew from the numerical evaluation is that the set of designed hand gestures as a means of interaction provides, for the combination of all tasks, similar user experience in comparison to a standard mouse device. Although the evaluation of both interfaces resulted in different grades for some particular tasks, the summation of ratings for all tasks in each of the interfaces results in close numbers. When considering hand gestures, such a summation results in 51 while, when considering the mouse device interface, it results in 52. Therefore, since the difference between results is of only one point, it is possible to state that both interaction styles, with their positive and negative aspects, provide likewise usability.

When analysing the results of the numerical evaluation from a lower-level perspective, taking into consideration groups of tasks with similar characteristics, a second conclusion can be drawn. Functionalities that require high accuracy or precision, such as the Simple Selection (Section 5.1.3) and Area Selection (Section 5.1.4) tasks, received lower ratings when executed through hand gestures in comparison to the mouse device. Contrarily, all other tasks—with a couple of exceptions that will be discussed— which do not require such level of accuracy, such as the Continuous Selection (Section 5.1.3) task, received equal or higher ratings for execution via hand gestures in comparison to the mouse device. The authors understand, according to the expert comments, that the levels of accuracy and precision are compromised in the developed gesture interface mainly because of two factors. The first one is that users might involuntarily experience hand tremors at some level, which makes it difficult for them to maintain the Leap Motion cursor still on a small target. The second factor involves the very anatomy of the human hand. As muscles, ligaments and tendons are connected, the contraction or relaxation of any of them has an impact on the rest. Thus, when users make the thumb folding motion in the selection task, the index finger ends up moving a little, which also alters the position of the Leap Motion cursor on the screen. Nevertheless, it is important to notice that the usability of the selection tasks is directly proportional to the size of the desired target over the Leap Motion cursor relation. The larger this relation grows, the less significant becomes the effects of accuracy issues, and the greater becomes the user experience. On the other hand, as nodes and edges become smaller and such a relation becomes closer to one, the greater becomes the importance of accuracy, and the more difficult it gets for users to select desired network elements.

When analysing tasks individually, besides the ones previously mentioned, interesting discussions arise. Figure 6.35 reveals that Relocate (Section 5.1.7) and Find Shortest Path (Section 5.1.8) are the only tasks that received higher ratings for execution through hand gestures in comparison to the mouse device. The expert comments do not provide deep insights on the reasons for such higher grades, but the authors assume that they simply reflect the expert's enthusiasm with the intuitiveness, ease of execution, ease to learn, and high recognition that such hand gestures produced against the usual interaction style enabled by standard mouse devices. Note that such gestures are not dependent on the accuracy or precision of the Leap Motion cursor, as they are not selection tasks.

Figure 6.35 also reveals that the Select Value (Section 5.1.6) task received a lower rating for execution using hand gestures in comparison to the standard mouse device. The expert comments together with the hand gestures recording provide insights that explain such a grade. The conclusion is that the hand gesture mapped to the Select Value task—the clockwise and anti-clockwise index finger spinning— was not unsuccessfully designed, but just not harmonious with the graphical user interface (GUI) composition. In the current layout, if a user desires to select a value located at the ending part of a long array of values, as when the expert was required to select the value "Sweden" from an alphabetically ordered list of countries, the user has to repeat the index finger spinning motion for several times, which becomes uncomfortable after some point. The problem is not with the spinning motion itself, but with its repetition. Therefore, the focus for the lower usability in this task shifts from the gesture itself to the choices regarding the GUI. Such a shift shows that the definition of a successful or unsuccessful hand gesture design has strong correlation with the graphical presentation of the user interface, and that it is the combination of both factors that determines the satisfaction of a user experiencing the application.

It is also worthy of discussing another factor which negatively affected the overall interaction experience with hand gestures that the expert underwent. This point was brought up in the expert comments section and does not regard any of the gestures in particular, but the user interface. The lack of a visual feedback feature capable of revealing to users the current status of the application, such as if a hand posture has been captured and recognised or which motion options are available at that moment, caused uncertainty to the expert and, ultimately, resulted in the interruption half-way through the execution of some task during the expert review. Such interruptions, as expected, reduced the expert's overall level of satisfaction.

Another interesting behaviour observed while the expert was testing the application, as written in the expert comments, was that the loading functionality—a feature that counts the number of consecutive frames of a specific hand posture—when the expert performed the Confirm Value (Section 5.1.6) task could never complete the necessary loading time to accomplish the task. The expert's Leap Motion Controller recognised the thumbs-up hand posture, but it always interrupted the loading cycle half-way through the counting. Similar behaviour was observed when the expert chose filter options, but with the loading functionality taking much longer to count the number of frames than in the developers' computer environment. During the experiment, after such loading functionality was removed, the expert's Leap Motion recognised the hand postures and the expert was able to complete the tasks normally. The conclusion, based on the authors' interpretation and assumptions, is that the processing power of the users' computer interferes in the speed that frames are processed and, consequently, affects the usability of tasks that are dependent on frame counting.

Finally, it is relevant to discuss handedness. The application and all functionalities it includes have been designed and developed focusing only on right-handed individuals. The reason behind such a choice is that both developers have the right hand as their dominant hand and also that most of the population is right-handed. Consequently, all hand postures and gestures were designed to be optimal for those individuals, but not for left-handed people. This matter did not come as an issue during the experimental phase as the expert who reviewed the application is also right-handed.

8 Conclusion

This project has investigated the use of hand gestures for the interactive analysis of two-dimensionally displayed multivariate networks in comparison to the standard interaction experience delivered through mouse devices. The work started with a literature review of the main fields that constitute the project background: information visualisation, HCI and graph drawing. This reviewing activity enlightened the authors with the fundamental concepts necessary for further carrying on the research work. Simultaneously, the work proceeded to its implementation phase, where the infrastructure required for the expert review activity was built and hand gestures were designed. Ultimately, the expert was contacted for the experimental phase and the obtained results discussed.

The discussion of the data collected from the expert review activity reveals that the developed gesture interface, including the proposed hand gestures and its graphical user interface, produces a similar user experience to the defined benchmark—the traditional interaction through mouse devices. However, the discussion also exposes that the set of tasks which involve the selection of network elements was impaired the most, mainly due to the accuracy difficulties to be overcome. Nevertheless, the authors understand such challenges as possibilities of improvement that, if achieved, can enhance the gesture interface's usability, even surpassing the benchmark score. This study has also shown that the usability of proposed hand gestures is strongly correlated to the manner in which the graphical user interface is structured. Thus, the overall user experience can also be improved by the harmonious arrangement of both factors.

The authors believe that the results presented in this study are relevant both to the industry, as they reveal that there is room for a shift without compromising the user experience in the traditional manner which people interact with software, as well as to academia, as they indicate that there are still many aspects to be further explored in this field of interest. Moreover, as mentioned in the first chapter, graph and networks are nowadays employed in many applications. Therefore, the authors believe that the discoveries, changes and improvements in the search for more natural and effortless ways of interaction—that this study has only started—have the potential to directly affect societies, which are more and more open to the advancement of new technologies. The results discussed indicate a positive impression and feedback towards touchless interfaces which, if further investigated and applied in other fields as it has already started been, can benefit several professionals by, for example, providing a safer work environment during medical and dental surgeries.

8.1 Future Work

Due to the limited resources within the scope of this degree project and the modification of the initial plans, there are various opportunities to continue investigating and developing the research work here started. The expert review activity revealed positive aspects of the designed gesture interface and application, but also provided interesting insights on other aspects that can be further improved or developed.

Starting with the design of hand gestures, which is the focus of this project, the further investigation of selection gestures that help users with the accuracy matter (e.g., gestures less susceptible to hand tremors or adjustment of Leap Motion cursor according to the size of nodes and edges) belongs to future work. Furthermore, the set of gestures can be even extended to accommodate more network analysis tasks, such as moving vertices in a network, set operation (e.g., union and intersection) on collections of vertices, find clusters, among other tasks mentioned in the first chapter. Additionally, hand gestures can be adapted to become optimal for left-handed individuals and enable them to also experience the developed interface. Furthermore, a new gesture calibration feature that enables users to adjust the implemented gestures to best match their natural hand postures before starting the analysis can be developed in the future. Another suggestion is to train the system to automatically detect unintended motion, such as hand tremors, which would help users to maintain precision. These features have the potential to increase usability by improving gesture recognition.

Other aspects that are not the gestures themselves can also be developed to improve the overall user experience, such as improvement of the algorithm and GUI. Data structures and searching algorithms utilised throughout the implementation are not optimal, which consequences can be observed through delays when processing complex operations and when handling large amounts of network elements. It was not an objective of this degree project to focus on algorithms and data structures employed, but an upgrade on those could end up benefiting the users' perception of the application. Also, the development of a visual feedback feature, mentioned in Chapter 7, showing the application status or what is recognised in the Leap Motion Controller field of view has the potential to increase the users' confidence on the gestures being executed.

Moreover, the initial plan of carrying out a study with several participants in which the execution performance of each task, both through hand gestures and a standard mouse device, are measured and analysed can be resumed in future. Such a study would improve the reliability and validity of the results by increasing the number of participants involved in the study—the sample size.

References

[1] Robert Koch Institute and Humboldt University of Berlin, “Route analysis & effective distance,” http://rocs.hu-berlin.de/corona/docs/analysis/visual_analytics/, [Online; accessed 17-February-2020].

[2] ——, “Graph visualisation tool,” http://rocs.hu-berlin.de/viz/sgb/, [Online; accessed 17-February-2020].

[3] S. van Den Elzen and J. J. van Wijk, “Multivariate network exploration and presentation: From detail to overview via selections and aggregations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2310–2319, 2014. [Online]. Available: https://doi.org/10.1109/TVCG.2014.2346441

[4] D. Rempel, M. J. Camilleri, and D. L. Lee, “The design of hand gestures for human–computer interaction: Lessons from sign language interpreters,” *International Journal of Human - Computer Studies*, vol. 72, no. 10–11, pp. 728–735, 2014. [Online]. Available: https://doi.org/10.1016/j.ijhcs.2014.05.003

[5] F. K. H. Quek, “Eyes in the interface,” *Image and Vision Computing*, vol. 13, no. 6, pp. 511–525, 1995. [Online]. Available: https://doi.org/10.1016/0262-8856(95)94384-C

[6] ——, “Toward a vision-based hand gesture interface,” in *Virtual Reality Software and Technology*, ser. VRST ’94. USA: World Scientific Publishing Co., Inc., 1994, p. 17–31. [Online]. Available: https://doi.org/10.1142/9789814350938_0003

[7] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn, “User-defined gestures for augmented reality,” in *Human-Computer Interaction – INTERACT 2013*, P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 282–299. [Online]. Available: https://doi.org/10.1007/978-3-642-40480-1_18

[8] Ultraleap, “Leap motion controller,” https://www.ultraleap.com/product/leap-motion-controller/, [Online; accessed 30-January-2020].

[9] Yi-Jheng Huang, T. Fujiwara, Yun-Xuan Lin, Wen-Chieh Lin, and Kwan-Liu Ma, “A gesture system for graph visualization in virtual reality environments,” in *2017 IEEE Pacific Visualization Symposium (PacificVis)*, vol. 1. IEEE, April 2017, pp. 41–45. [Online]. Available: https://doi.org/10.1109/PACIFICVIS.2017.8031577

[10] K. Burshtein, T. Jones, and N. Smith. (2013) Exploring twitter networks with the leap motion student project. [Online]. Available: http://jonmccormack.info/wp-content/uploads/2013/11/FinalReport.pdf

[11] Leap Motion, Inc., “Tracking Model - Leap Motion Javascript SDK,” https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Tracking.html, [Online; accessed 17-March-2020].

[12] ——, “API Overview - Leap Motion Javascript SDK,” https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html, [Online; accessed 17-March-2020].

[13] F. R. Giordano, W. P. Fox, S. B. Horton, and M. D. Weir, *A first course in mathematical modeling*, 4th ed. Pacific Grove, Calif. : London: Brooks/Cole ; Cengage Learning [distributor], 2008.

[14] S. Pirzada, “Applications of graph theory,” *PAMM: Proceedings in Applied Mathematics and Mechanics*, vol. 7, no. 1, pp. 2 070 013–2 070 013, 2007. [Online]. Available: https://doi.org/10.1002/pamm.200700981

[15] O. Mason and M. Verwoerd, "Graph theory and networks in biology," *IET Systems Biology*, vol. 1, no. 2, pp. 89–119, 2007. [Online]. Available: https://doi.org/10.1049/iet-syb:20060038

[16] K. Kaundal, "Applications of graph theory in everyday life and technology," *Imperial Journal of Interdisciplinary Research (IJIR)*, vol. 3, no. 3, pp. 892–894, 2017.

[17] I. Jusufi, "Multivariate networks: visualization and interaction techniques," Ph.D. dissertation, Linnaeus University, Faculty of Engineering and Technology, Department of Computer Science, Växjö, 2013.

[18] I. Jusufi, Y. Dingjie, and A. Kerren, "The network lens: Interactive exploration of multivariate networks using visual filtering," in *2010 14th International Conference Information Visualisation*. IEEE, July 2010, pp. 35–42. [Online]. Available: https://doi.org/10.1109/IV.2010.15

[19] E. Banissi, F. T. Marchese, C. Forsell, and J. Johansson, Eds., *Information visualisation: techniques, usability and evaluation*. Newcastle upon Tyne: Cambridge Scholars Publishing, 2014.

[20] A. Kerren, H. C. Purchase, and M. O. Ward, Eds., *Multivariate network visualization : Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised discussions*, ser. Lecture notes in computer science; 8380. Cham: Springer, 2014.

[21] B. A. Myers, "A brief history of human-computer interaction technology," *Interactions*, vol. 5, no. 2, pp. 44–54, 1998. [Online]. Available: https://doi.org/10.1145/274430.274436

[22] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale, "Beyond mouse and keyboard: Expanding design considerations for information visualization interactions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2689–2698, 2012. [Online]. Available: https://doi.org/10.1109/TVCG.2012.204

[23] B. Zimmer, "Guided interaction and collaborative exploration in heterogeneous network visualizations," Ph.D. dissertation, Linnaeus University, Faculty of Technology, Department of Computer Science and Media Technology, Växjö, 2019.

[24] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* IEEE, 2005, pp. 111–117. [Online]. Available: https://doi.org/10.1109/INFVIS.2005.1532136

[25] B. Lee, C. Plaisant, C. Sims, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization," in *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, ACM. Venezia, Italy: ACM, May 2006, pp. 1–5. [Online]. Available: https://hal.inria.fr/hal-00851754

[26] A. Kerren. (2020) Information visualisation. Lecture slides. [Online]. Available: https://cs.lnu.se/isovis/courses/spring20/4dv800/lessons/01-intro.pdf

[27] C. Ware, *Information visualization : perception for design*, 3rd ed., ser. Interactive Technologies. Waltham, MA 02451, USA: Elsevier/MK, 2013.

[28] J. S. Yi, Y. a. Kang, J. Stasko, and J. A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–1231, Nov 2007. [Online]. Available: https://doi.org/10.1109/TVCG.2007.70515

[29] W. Fikkert, M. D'Ambros, T. Bierz, and T. J. Jankun-Kelly, *Interacting with Visualizations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 77–162. [Online]. Available: https://doi.org/10.1007/978-3-540-71949-6_3

[30] A. Kerren and F. Schreiber, “Toward the role of interaction in visual analytics,” in *Proceedings of the Winter Simulation Conference*, ser. WSC ’12, no. 420. IEEE, 2012, pp. 1–13. [Online]. Available: https://doi.org/10.1109/WSC.2012.6465208

[31] M. Wybrow, B. Zimmer, N. Elmqvist, J.-D. Fekete, T. Von Landesberger, and J. Van Wijk, “Interaction in the Visualization of Multivariate Networks,” in *Multivariate Network Visualization*, ser. Lecture Notes in Computer Science, A. Kerren, H. Purchase, and M. O. Ward, Eds. Springer, Apr. 2014, vol. 8380, pp. XV, 237. [Online]. Available: https://hal.inria.fr/hal-00974335

[32] The Interaction Design Foundation, “What is Human-Computer Interaction (HCI),” https://www.interaction-design.org/literature/topics/human-computer-interaction, [Online; accessed 23-February-2020].

[33] W. Büschel, J. Chen, R. Dachselt, S. Drucker, T. Dwyer, C. Görg, T. Isenberg, A. Kerren, C. North, and W. Stuerzlinger, *Interaction for Immersive Analytics*. Cham: Springer International Publishing, 2018, pp. 95–138. [Online]. Available: https://doi.org/10.1007/978-3-030-01388-2_4

[34] G. R. S. Murthy and R. S. Jadon, “A review of vision based hand gestures recognition,” *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009. [Online]. Available: http://csjournals.com/IJITKM/PDF/34-G.R.S.Murthy_R.S.Jadon.pdf

[35] A. Sharma, A. Yadav, S. Srivastava, and R. Gupta, “Analysis of movement and gesture recognition using leap motion controller,” *Procedia Computer Science*, vol. 132, pp. 551–556, 2018, International Conference on Computational Intelligence and Data Science. [Online]. Available: https://doi.org/10.1016/j.procs.2018.05.008

[36] J. Phung, “Leap Motion Teardown in Macro HD,” (July 27, 2013), [Video File]. [Online]. Available: https://www.youtube.com/watch?v=uF0NSUmxFYA

[37] Ultrahaptics Ltd. (2019, Dec.) Leap motion controller data sheet. UH-003206-TC Issue 2. [Online]. Available: https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf

[38] Ultraleap, “Tracking, how it works,” https://www.ultraleap.com/tracking/#how-it-works, [Online; accessed 19-February-2020].

[39] G. M. Rosa and M. L. Elizondo, “Use of a gesture user interface as a touchless image navigation system in dental surgery: Case series report,” *Imaging science in dentistry*, vol. 44, no. 2, pp. 155–160, 2014. [Online]. Available: https://doi.org/10.5624/isd.2014.44.2.155

[40] Wikipedia contributors, “Force-directed graph drawing — Wikipedia, the free encyclopedia,” https://en.wikipedia.org/w/index.php?title=Force-directed_graph_drawing&oldid=944695567, 2020, [Online; accessed 8-April-2020].

[41] T. Fujiwara, J. Chou, A. M. McCullough, C. Ranganath, and K. Ma, “A visual analytics system for brain functional connectivity comparison across individuals, groups, and time points,” in *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2017, pp. 250–259. [Online]. Available: https://doi.org/10.1109/PACIFICVIS.2017.8031601

[42] T. Freeman, L. Goldovsky, M. Brosch, S. Van Dongen, P. Maziere, R. Grocock, S. Freilich, J. Thornton, and A. Enright, “Construction, visualisation, and clustering of transcription networks from microarray expression data,” *PLOS Computational Biology*, vol. 3, no. 10, pp. 1–11, 10 2007. [Online]. Available: https://doi.org/10.1371/journal.pcbi.0030206

[43] K. Burshtein, T. Jones, and N. Smith, “Twitter Network Visualisation with the Leap Motion,” (November 19, 2013), [Video File]. [Online]. Available: https://vimeo.com/79839419

[44] J. Hagelbäck. (2019) Method overview - degree projects in computer science. [Online]. Available: https://coursepress.lnu.se/subject/thesis-projects/method-overview/

[45] ——. (2017) Verification and validation - degree projects in computer science. [Online]. Available: http://coursepress.lnu.se/subject/thesis-projects/verification-and-validation/

[46] A. T. Bahill and S. J. Henderson, “Requirements development, verification, and validation exhibited in famous failures,” *Systems Engineering*, vol. 8, no. 1, pp. 1–14, 2005. [Online]. Available: https://doi.org/10.1002/sys.20017

[47] B. Kitchenham, “Procedures for performing systematic reviews,” Keele University, Tech. Rep. TR/SE-0401, 07 2004. [Online]. Available: http://www.inf.ufsc.br/~aldo.vw/kitchenham.pdf

[48] M. Yue, K. Tian, and T. Ma, “An accurate and impartial expert assignment method for scientific project review,” *Journal of Data and Information Science*, vol. 2, no. 4, pp. 65–80, 2017. [Online]. Available: https://doi.org/10.1515/jdis-2017-0020

[49] U.S. Department of Health & Human Services, “Heuristic evaluations and expert reviews,” https://www.usability.gov/how-to-and-tools/methods/heuristic-evaluation.html, [Online; accessed 03-April-2020].

[50] M. Tory and T. Möller, “Evaluating visualizations: do expert reviews work?” *IEEE Computer Graphics and Applications*, vol. 25, no. 5, pp. 8–11, 2005. [Online]. Available: https://doi.org/10.1109/MCG.2005.102

[51] J. Wilson, *Essentials of Business Research: A Guide to Doing Your Research Project*. SAGE Publications, 2010.

[52] E. R. Babbie, *The Practice of Social Research*, 12th ed. Belmont, CA: USA: Wadsworth ; Cengage Learning [distributor], 2010.

[53] J. Hagelbäck. (2017) Validity - degree projects in computer science. [Online]. Available: https://coursepress.lnu.se/subject/thesis-projects/validity/

[54] L. Cohen, L. Manion, and K. Morrison, *Research methods in education*, 6th ed. Milton Park, Abingdon, Oxon, [England] ; New York: Routledge, 2007.

[55] C. Tessier and V. Bonnemains, “Experiments with participants: Some ethical considerations,” in *Neuroergonomics*, H. Ayaz and F. Dehais, Eds. Academic Press, 2019, ch. 11, pp. 65–70. [Online]. Available: https://doi.org/10.1016/B978-0-12-811926-6.00011-7

[56] J. Leitet. (2019) Client-based web programming: Spa and offline. [Online]. Available: https://coursepress.lnu.se/kurs/klientbaserad-webbprogrammering/lectures/l06-offline-spa/

[57] Vue.js, “Vue.js,” https://vuejs.org, [Online; accessed 8-April-2020].

[58] Ember.js, “Ember.js - a framework for ambitious web developers,” https://emberjs.com, [Online; accessed 8-April-2020].

[59] Google, “Angularjs — superheroic javascript mvw framework,” https://angularjs.org, [Online; accessed 8-April-2020].

[60] Facebook, “React,” https://reactjs.org/, [Online; accessed 01-February-2020].

[61] World Wide Web Consortium. (Jan. 19, 2005) Document Object Model (dom). [Online]. Available: https://www.w3.org/DOM/

[62] Google, “AngularJS: Developer Guide: Introduction,” https://docs.angularjs.org/guide/introduction, [Online; accessed 01-February-2020].

[63] C. Harrington, “React vs AngularJS vs KnockoutJS: a Performance Comparison,” https://www.codementor.io/@chrisharrington/react-vs-angularjs-vs-knockoutjs-a-performance-comparison-85hwzepbz, [Online; accessed 09-March-2020].

[64] Facebook, “Rendering Elements - React,” https://reactjs.org/docs/rendering-elements.html, [Online; accessed 09-March-2020].

[65] C. A. Miller, J. Anthony, M. M. Meyer, and G. Marth, “Scribl: an HTML5 Canvas-based graphics library for visualizing genomic data over the web,” *Bioinformatics*, vol. 29, no. 3, pp. 381–383, 2013. [Online]. Available: https://doi.org/10.1093/bioinformatics/bts677

[66] Leap Motion, Inc., “Connecting to the Controller - Leap Motion Javascript SDK,” https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Controllers.html, [Online; accessed 11-March-2020].

[67] ——, “WebSocket Communication - Leap Motion Javascript SDK,” https://developer-archive.leapmotion.com/documentation/javascript/supplements/Leap_JSON.html, [Online; accessed 11-March-2020].

[68] ——, “Frames - Leap Motion Javascript SDK,” https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Frames.html#id482, [Online; accessed 17-March-2020].

[69] J. Lin, C. Harris-Adamson, and D. Rempel, “The design of hand gestures for selecting virtual objects,” *International Journal of Human–Computer Interaction*, vol. 35, no. 18, pp. 1729–1735, 2019. [Online]. Available: https://doi.org/10.1080/10447318.2019.1571783

[70] Zoom Video Communications, Inc., “Video Conferencing, Web Conferencing, Webinars, Screen Sharing - Zoom,” https://zoom.us, [Online; accessed 24-April-2020].

A Questionnaire

General Instructions:

Before performing the evaluation tasks presented in this questionnaire, it is recommended that you familiarise yourself with the user interface, functionalities available and their corresponding hand gestures by training in the play mode. It is also suggested that you use the play mode to find a position for the Leap Motion Controller device that achieves high recognition of hand postures and enables you to perform gestures with ease. When you feel ready to start the evaluation, conduct the entire sequence of tasks using a standard mouse device and then repeat the process using the developed gesture interface. Please, take concise and objective notes after the completion of each task. **It would be of great value to have (1) comments about your experience, (2) answers for the questions, and (3) a rating from 1 (horrible) to 5 (excellent) on your overall experience.** Such notes should support your judgment of both interaction approaches (overall and specific functionalities) and enable us to draw conclusions from your review.

Tasks:

1. Using only panning, zooming and hoovering, find the disconnected node (no edges) in the network and provide its id number and first name.

   ....
   ....
   ....

2. Using the simple selection functionality, select node 25, read its attributes on the left sidebar, and provide its gender.

   ....
   ....
   ....

3. Using the continuous selection functionality, select the edge between nodes 22 and 31 and provide its relationship type available from the left sidebar.

   ....
   ....
   ....

4. Deselect all previously selected network elements (nodes and edges).

   ....
   ....
   ....

5. Using the area selection functionality, select nodes 8, 10, 22, and 35.

   ....
   ....
   ....

6. Using the find adjacent nodes functionality, find all nodes that are either directly connected (first iteration) or connected by an intermediate node (second iteration) to the group of nodes selected in task 5.

   ....
   ....
   ....

7. Using the filter functionality, find all node(s) that have Sweden as value for the attribute country and provide its/their id numbers.

   ....
   ....
   ....

8. Using the selection and find shortest path functionalities, discover the shortest path between node 9 and node 2, and provide the length of the path (count hops).

   ....
   ....
   ....

9. Using the relocate functionality, revisit the result of task 6.

   ....
   ....
   ....

10. Using the area selection and filter functionalities, select all connected nodes in the network and find all nodes that have bigger or equal to 29 as value for the attribute age.

    ....
    ....
    ....

Please, use the space below for providing positive comments, feedback regarding the overall experience not related to a specific task, reporting features that did not work as expected, or bringing up to our attention any other issues that are relevant for you.