

# **CONTRIBUTIONS TO THE VISUALIZATION PROCESS AND MANIPULATION OF PSEUDO- HOLOGRAPHIC IMAGES**

Teză destinată obținerii  
titlului științific de doctor inginer  
la  
Universitatea Politehnica Timișoara  
în domeniul CALCULATOARE ȘI TEHNOLOGIA  
INFORMAȚIEI  
de către

**Inf. Ioan Virag**

Conducător științific: prof.univ.dr.ing. Lăcrămioara Stoicu-Tivadar  
Referenți științifici: prof.univ.dr.  
prof.univ.dr.ing.  
prof.univ.dr.

Ziua susținerii tezei: .2017

Seriile Teze de doctorat ale UPT sunt:

- |   |  |
|---|--|
| 1. Automatică                               | 9. Inginerie Mecanică                      |
| 2. Chimie                                   | 10. Știința Calculatoarelor                |
| 3. Energetică                               | 11. Știința și Ingineria Materialelor      |
| 4. Ingineria Chimică                        | 12. Ingineria sistemelor                   |
| 5. Inginerie Civilă                         | 13. Inginerie energetică                   |
| 6. Inginerie Electrică                      | 14. Calculatoare și tehnologia informației |
| 7. Inginerie Electronică și Telecomunicații | 15. Ingineria materialelor                 |
| 8. Inginerie Industrială                    | 16. Inginerie și Management                |

Universitatea Politehnica Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul Școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2017

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității Politehnica Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,  
Tel./fax 0256 403823  
e-mail: editura@edipol.upt.ro

## Foreword

The thesis was elaborated during my activity in the Department of Automation and Applied Informatics of Politehnica University of Timișoara.

As an alternative method to the classical command-line interfaces (CLIs) and later graphical user interfaces (GUIs), new ways of multimodal human-computer interaction (HCI) emerged in the latest decades introducing several options known as natural user interfaces (NUIs). The key improvement is that these interfaces allow the user to feel in control even not mastering yet the entire spectrum that the underlying technology has to offer in a particular computing system. This natural way of interaction with a machine provides an easier choice to learn when the targeted users have various degrees of computer skills because of the interaction modalities involved relying on gesture tracking, voice recognition and in some cases reading directly the electrical signals from the brain.

One of the fundamental aspects to be considered in biomedical research is to improve the underlying software and hardware that makes it possible to interact with the studied three-dimensional images by using gestures. The herein thesis addresses these problems and provides solutions based on the latest open source software adapted to the next generation of input and output devices. Starting with an overview of the latest technologies, the paper presents original modeling and implementation using gesture tracking and pseudo-holographic 3D visualization devices, which are validated by the usability evaluation that mirrors the user performance and user-satisfaction levels in conjunction with the proposed configurations.

The presented solutions have a large impact area covering medicine, education, architecture and various fields of engineering where complex data sets need to be visualized on larger displays and the understanding of these cases is facilitated by enabling gesture based interaction with the rendered 3D models without any prior calibration of the input device in use.

This study and results will provide a valuable reference point for future research, in cases where image analysis has a key role for understanding complex problems that need to be visualized using the next generation of autostereoscopic display technologies.

Timișoara, 09.2017

Virag Ioan

I would like to express my gratitude to my PhD advisor, prof. dr. eng. Lăcrămioara Stoicu-Tivadar, for her continuous support and guidance through my research, for her patience when I was facing specific problems and her most valuable advice that helped me surpass them.

My sincere thanks go to my colleagues and collaborators, especially to prof. dr. eng. Diana Lungeanu, assoc. prof. dr. eng. Mihai Medeleanu and assist. prof. dr. eng. Mihaela Crisan-Vida for their aid in all the aspects of my research.

Special thanks goes to my family, my little daughter Maia and my wife Bogdana, who supported and encouraged me to go further when I was lacking inspiration.

This paper is partially supported by the Sectoral Operational Programme Human Resources Development POSDRU/159/1.5/S/137516 financed from the European Social Fund and by the Romanian Government.

Virag, Ioan

**Contributions to the visualization process and manipulation of pseudo-holographic images**

Teze de doctorat ale UPT, Seria X, Nr. YY, Editura Politehnica, 2017, 166 pagini, 102 figuri, 22 tabele.

Cuvinte cheie:

medical visualization system, medical image analysis, image segmentation, gesture tracking

Rezumat,

The thesis contributes to the development of new algorithms, models and applications for medical visualization systems based on the last generation of input and output hardware devices. These components have an important role in biomedical research, medical training, patient diagnosis, surgery planning and treatment monitoring. From the technological point of view the solutions presented in the thesis focus on the gesture based interaction with the rendered 3D models that can be visualized using an autostereoscopic display. This requires a comprehensive analysis of the current software and hardware technologies involved in three-dimensional pseudo-holographic imaging. Also a key aspect is that all the software is browser-based eliminating the need for any specific operating system that in some cases are a barrier for nontechnical users. Due to the limitations imposed by the nature of the involved software technologies a new segmentation algorithm was developed and implemented that allows a faster execution. The software tools were used on publicly available real medical image sets and 3D molecular models in order to validate the results based on a proposed methodology and specific test scenarios for students in medicine and chemistry

## CONTENTS

Foreword .....	3
LIST OF ACRONYMS .....	7
LIST OF FIGURES.....	9
LIST OF TABLES .....	12
1. INTRODUCTION.....	13
1.1. Introduction into the research topic.....	13
1.2. Objectives of the PhD thesis .....	14
1.3. Structure of the thesis .....	15
2. HARDWARE TECHNOLOGIES OVERVIEW .....	17
2.1. Input devices based on gesture recognition .....	17
2.1.1. Leap Motion.....	18
2.1.2. MYO Armband.....	19
2.1.3. SideSwipe .....	21
2.1.4. SoundWave.....	22
2.1.5. Project Soli.....	23
2.2. Brain-computer interfaces .....	25
2.2.1. EPOC Plus headset .....	25
2.2.2. Functional Electrical Stimulation.....	27
2.2.3. Osseointegrated human-machine gateway .....	28
2.2.4. BrainGate.....	30
2.3. 3D Display technologies .....	31
2.3.1. Stereoscopic displays.....	33
2.3.2. Autostereoscopic displays.....	34
2.3.3. Volumetric 3D displays .....	35
2.3.4. Pseudo-holographic displays.....	37
2.3.5. Holographic film projections .....	39
2.3.6. Heliodisplay.....	40
2.3.7. Braille terminals.....	41
2.4. Haptic devices .....	42
2.5. Conclusions .....	44
3. SOFTWARE TECHNOLOGIES. OVERVIEW AND CONTRIBUTIONS .....	50
3.1. NanoDICOM .....	51
3.2. JSARToolKit.....	52
3.3. CornerstoneJS .....	54
3.4. HTML5 .....	55
3.5. WebRTC.....	58
3.6. WebGL.....	59
3.7. ThreeJS .....	59
3.8. LeapJS.....	60
3.9. ThreeLeapControls .....	61
3.10. Conclusions .....	62
4. DEVELOPMENT OF A BROWSER-BASED DICOM VIEWER .....	65
4.1. Application architecture and software modules.....	67
4.2. Image segmentation algorithm for restrictive environments .....	70
4.3. Image colorization.....	73
4.4. Multiplanar 3D reconstruction .....	76
4.5. Loading the 2D slices.....	79
4.6. Controlling the 3D model with a LEAP Motion Controller .....	81
4.7. Conclusions .....	84

5. INTERACTING WITH A 3D MOLECULAR VIEWER USING GESTURES .....	85
5.1. The GImol 3D molecular viewer.....	85
5.2. Adapting the application to gesture-based interaction .....	88
5.3. Conclusions .....	92
6. EVALUATION .....	93
6.1. Usability evaluation overview for the JSDV application .....	94
6.2. Methodology for testing the JSDV application.....	96
6.2.1. Participants .....	97
6.2.2. Training .....	98
6.2.3. Procedure .....	98
6.3. Test results for the JSDV application.....	99
6.4. Usability evaluation overview for the GLmol LEAP application.....	112
6.5. Methodology for testing the GLmol LEAP application .....	113
6.5.1. Participants .....	114
6.5.2. Training .....	115
6.5.3. Procedure .....	115
6.6. Test results for the GLmol LEAP application .....	116
6.7. Conclusions .....	127
7. CONCLUSIONS AND FURTHER DEVELOPMENT .....	128
ANNEX 1.....	132
ANNEX 2.....	145
REFERENCES.....	158

## LIST OF ACRONYMS

2D	Two-dimensional
3D	Three-dimensional
ALS	Amyotrophic Lateral Sclerosis
API	Application Programming Interface
BANA	Braille Authority of North America
BCI	Brain-Computer Interface
CE	Consumed Endurance
CLI	Command-Line Interface
CNS	Central Nervous System
CRT	Cathode Ray Tube
CSS	Cascading Style Sheets
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DNI	Direct Neural Interface
DOC	Disorders of Consciousness
DSSS	Direct-Sequence Spread Spectrum
ECoG	Electrocorticography
EEG	Electroencephalography
EMG	Electromyography
FES	Functional Electrical Stimulation
FMCW	Frequency Modulated Continuous Wave
fMRI	Functional Magnetic Resonance Imaging
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HID	Human Interface Device
HMI	Human-Machine Interaction
HTML	HyperText Markup Language
IETF	Internet Engineering Task Force
IMU	Inertial Measurement Unit
ISM	Industrial, scientific and medical
JSARToolKit	JavaScript Augmented Reality Toolkit
LCD	Liquid crystal display
LED	Light-emitting diode
LIS	Locked-In Syndrome
LoS	Line of Sight
MCS	Minimally Conscious State
MEG	Magnetoencephalography
MPL	Modular Prosthetic Limb
MPR	Multi-planar reconstruction
MRI	Magnetic Resonance Imaging
NCBI	National Center for Biotechnology Information
NEMA	National Electrical Manufacturers Association
NUI	Natural User Interface

OHMG	Osseointegrated human-machine gateway
OPRA	Osseointegrated Protheses for the Rehabilitation of Amputees
PACS	Picture Archiving and Communication System
PCB	Printed Circuit Board
PDB	Protein Data Bank
PET	Positron Emission Tomography
PNG	Portable Network Graphics
RCSB	Research Collaboratory for Structural Bioinformatics
SCI	Spinal Cord Injury
SDF	Structure Data Format
TLD	Tracking Learning Detection
TMR	Targeted Muscle Reinnervation
TOF	Time of Flight
USB	Universal Serial Bus
VS	Vegetative State
W3C	World Wide Web Consortium
WADO	Web Access to DICOM Objects
WebRTC	Web Real-Time Communication
WebGL	Web Graphics Library
WHATWG	Web Hypertext Application Technology Working Group



## LIST OF FIGURES

Figure 1.1. Structure of the thesis.....	16
Figure 2.1. Leap Motion interaction space .....	18
Figure 2.2. Leap Motion hardware .....	18
Figure 2.3. Objects illuminated by the infrared LEDs .....	19
Figure 2.4. MYO detecting electrical activity in muscles.....	20
Figure 2.5. Two MYO armbands used to controll a modular prosthetic limb.....	20
Figure 2.6. The SideSwipe system after [32] .....	21
Figure 2.7. SoundWave in-air gesture sensing after [33] .....	22
Figure 2.8. AirLink interaction between multiple devices after [35].....	23
Figure 2.9. Soli converting gestures into digital signals .....	24
Figure 2.10. Soli chip and antennas .....	24
Figure 2.11. EPOC headset.....	26
Figure 2.12. Electrode array implanted next to the spinal cord and a pulse generator below .....	27
Figure 2.13. Patient performing FES cycling .....	28
Figure 2.14. The connection of the OHMG system to the peripherall nerves after [46] .....	29
Figure 2.15. Patient performing his daily activities with the OHMG system after [46] .....	29
Figure 2.16. Patient drinking from a bottle using BrainGate after [48] .....	30
Figure 2.17. Clarity Matrix 3D display by Planar .....	33
Figure 2.18. Operating principle of a parallax-barrier vs. a lenticular lens.....	34
Figure 2.19. A static screen volumetric 3D display formed by RGB LEDs .....	35
Figure 2.20. Volumetric display using a moving projection screen .....	36
Figure 2.21. Aerial Burton 3D display .....	37
Figure 2.22. Pepper’s ghost technique used in theaters .....	38
Figure 2.23. VisPod 73D (HoloVision) pseudoholographic display by ProVision .....	38
Figure 2.24. DREAMOC HD3 pseudoholographic display by Realfiction .....	39
Figure 2.25. Holographic film projection.....	40
Figure 2.26. Heliodisplay .....	40
Figure 2.27. Braille terminal .....	41
Figure 2.28. Tactisplay displaying graphic content.....	42
Figure 2.29. Selman Surgery Rehearsal Platform .....	43
Figure 2.30. ScanTrainer .....	43
Figure 3.1. The virtually attached 3D model rotates in the same time as the real patient’s arm presented by the author in [73] .....	50
Figure 3.2. Example of an ID marker .....	53
Figure 3.3. The IR lights of the LEAP Motion Controller .....	60
Figure 4.1. Cornerstone example application that displays a DICOM file from the local file system.....	65
Figure 4.2. JSDV key functional modules .....	66
Figure 4.3. JSDV user interface.....	67
Figure 4.4. Web technologies used by JSDV .....	68
Figure 4.5. Interaction between the user and the various modules of JSDV .....	69
Figure 4.6. JSDV use case for interacting with the 3D model .....	70

Figure 4.7. The initial DICOM image on the left and the segmented image on the right.....	71
Figure 4.8. The segmentation process diagram presented by the author in [99] ...	72
Figure 4.9. The segmentation threshold set to zero .....	73
Figure 4.10. The segmented image on the left and the colorized image on the right .....	74
Figure 4.11. The bone tissue is isolated and colorized in the image on the right....	75
Figure 4.12. JSDV image colorization process .....	76
Figure 4.13. A 3D model rendered based on the colorized slices .....	77
Figure 4.14. JSDV multiplanar 3D reconstruction .....	78
Figure 4.15. Difference between the initial input element on the left and the pseudo-button on the right.....	79
Figure 4.16. A 2D plane from the 3D stack that uses as texture the image on the left .....	80
Figure 4.17. The process of loading images as textures .....	80
Figure 4.18. The cursor is shown when one finger and one hand is detected .....	82
Figure 4.19. The opacity of the selected slice changes based on the number of hands and fingers detected .....	83
Figure 4.20. The sequence of controlling the 3D model with gestures .....	83
Figure 5.1. GLmol user interface .....	85
Figure 5.2. GLmol display options .....	86
Figure 5.3. Interaction between the user and the GLmol application .....	87
Figure 5.4. GLmol LEAP main functionalities .....	88
Figure 5.5. Web technologies used for GLmol LEAP .....	89
Figure 5.6. Interaction between the modules of the GLmol LEAP application.....	90
Figure 5.7. Testing the GLmol application on a HoloVision display .....	91
Figure 5.8. GLmol LEAP use case for interacting with the 3D model.....	91
Figure 6.1. Custom-made pseudo-holographic 3D display .....	94
Figure 6.2. Setup used for testing the 3D user interface of the JSDV application ...	96
Figure 6.3. JSDV users age distribution.....	97
Figure 6.4. A small group of participants during the training phase .....	98
Figure 6.5. JSDV average completion rates .....	104
Figure 6.6. JSDV average error-free rates.....	104
Figure 6.7. JSDV Usability metrics.....	106
Figure 6.8. JSDV 2D mouse assessment .....	107
Figure 6.9. JSDV LEAP Motion assessment .....	109
Figure 6.10. JSDV 2D mouse vs. LEAP Motion assessment .....	111
Figure 6.11. Setup used for testing the 3D user interface of the GLmol LEAP application .....	113
Figure 6.12. GLmol LEAP users age distribution .....	114
Figure 6.13. GLmol LEAP average completion rates .....	119
Figure 6.14. GLmol LEAP average error-free rates.....	120
Figure 6.15. GLmol LEAP Usability metrics .....	121
Figure 6.16. GLmol LEAP 2D mouse assessment .....	123
Figure 6.17. GLmol LEAP application LEAP Motion assessment .....	124
Figure 6.18. GLmol LEAP 2D mouse vs. LEAP Motion assessment.....	126
Figure A.1. Consent form for the JSDV application .....	132
Figure A.2. Demographic information form for the JSDV application .....	133
Figure A.3. Task Scenarios for the JSDV application .....	134
Figure A.4. Time on task table for the JSDV application .....	135
Figure A.5. Usability Questionnaire for the JSDV application .....	136

Figure A.6. 2D Mouse Device Assessment Questionnaire for the JSDV application	138
Figure A.7. LEAP Device Assessment Assessment Questionnaire for the JSDV application	140
Figure A.8. Raw test results for annex 2 for the JSDV application	141
Figure A.9. Raw test results for annex 5 for the JSDV application	142
Figure A.10. Raw test results for annex 6 for the JSDV application	143
Figure A.11. Raw test results for annex 7 for the JSDV application	144
Figure A2.1. Consent form for the GLmol LEAP application	145
Figure A2.2. Demographic information form for the GLmol LEAP application	146
Figure A2.3. Task Scenarios for the GLmol LEAP application	147
Figure A2.4. Time on task table for the GLmol LEAP application	148
Figure A2.5. Usability Questionnaire for the GLmol LEAP application	149
Figure A2.6. 2D Mouse Device Assessment Questionnaire for the GLmol LEAP application	151
Figure A2.7. LEAP Device Assessment Questionnaire for the GLmol LEAP application	153
Figure A2.8. Raw test results for annex 2 for the GLmol LEAP application	154
Figure A2.9. Raw test results for annex 5 for the GLmol LEAP application	155
Figure A2.10. Raw test results for annex 6 for the GLmol LEAP application	156
Figure A2.11. Raw test results for annex 7 for the GLmol LEAP application	157

## LIST OF TABLES

Table 2.1 Hardware technologies comparison .....	45
Table 3.1 Supported Transfer Syntaxes by NanoDICOM .....	51
Table 3.2 The steps required to obtain a 3D image.....	52
Table 3.3 Supported Transfer Syntaxes by cornerstonWADOImageLoader.....	54
Table 3.4 Important new elements introduced in HTML5 .....	56
Table 3.5 Obsolete elements in HTML5 .....	57
Table 3.6 Software technologies comparison .....	62
Table 5.1 The approximate HU values of various tissue types .....	74
Table 6.1 JSDV time on task results showing the reference, average, minimum, maximum times and standard deviation .....	99
Table 6.2 JSDV average times on task, corrected sample variances, sample standard deviations and t-test results for the two user groups .....	100
Table 6.3 Recorded times per task scenarios for JSDV .....	101
Table 6.4 Completion rates and error-free rates for JSDV.....	103
Table 6.5 JSDV 2D mouse assesment questions, scale, averages and standard deviation. ....	106
Table 6.6 JSDV LEAP Motion assesment questions, scale, averages and standard deviation. ....	108
Table 6.7 Comparisson of the device assesment results for 2D mouse and LEAP Motion for JSDV .....	109
Table 6.8 GLmol LEAP time on task results showing the reference, average, minimum, maximum times and standard deviation .....	116
Table 6.9 GLmol LEAP average times on task, corrected sample variances, sample standard deviations and t-test results for the two user groups.....	116
Table 6.10 Recorded times per task scenarios for GLmol LEAP.....	117
Table 6.11 Completion rates and error-free rates for GLmol LEAP .....	118
Table 6.12 GLmol LEAP 2D mouse assesment questions, scale, averages and standard deviation. ....	122
Table 6.13 GLmol LEAP application LEAP Motion assesment questions, scale, averages and standard deviation. ....	123
Table 6.14 Comparisson of the device assesment results for 2D mouse and LEAP Motion for GLmol LEAP .....	125

# 1. INTRODUCTION

## 1.1. Introduction into the research topic

Medical visualization systems have a decisive role for physicians in establishing the right diagnosis and treatment. In the latest decades several solutions have been proposed with the purpose to improve these medical imaging technologies like introducing PACS (Picture Archiving and Communication Systems) [1] and the DICOM (Digital Imaging and Communications in Medicine) standard [2] as a registered trademark of NEMA (National Electrical Manufacturers Association). The direct result of these changes was the increased interoperability between different institutions and medical equipments provided by various manufacturers. However a major problem the users are facing when installing some of the applications is the dependence on other proprietary software components or operating systems. In order to solve these problems several web based solutions emerged [3] that allow the visualization of the image data encapsulated in the DICOM files inside a browser window. These web applications, based on open source software libraries and APIs (Application Programming Interfaces) allow their further development based on the introduction of new possibilities offered by the improvement of their modules over time.

Another important aspect related to medical image visualization is their use in operation planning or even during medical interventions. Accessing the images by conventional input devices, such as keyboard and mouse, would represent a major risk factor in operating rooms. In order to avoid such possible complications several systems based on gesture tracking have been proposed [4], [5], and [6] that allow the interaction with the images without any physical contact.

The web based applications can be adapted to mobile devices [7], [8] allowing specialist from different medical fields to collaborate, even if they are on different continents, with the direct result of increased accuracy of the investigation.

The image data in the patients' DICOM files obtained by different imaging techniques such as CT (Computed Tomography), MRI (Magnetic Resonance Imaging) or PET (Positron Emission Tomography) are stored as a group of 2D (two-dimensional) parallel slices representing the scanned body area that allow its 3D (three-dimensional) volume rendering using MPR (multi-planar reconstruction) [9]. These 3D images can be visualized on autostereoscopic (pseudo-holographic) displays without wearing any special glasses, hugely improving the collaboration between different teams of specialist [10].

During the research several approaches to build a browser-based medical visualization system were proposed, designed, implemented and tested that will be discussed in the thesis.

In a first theoretical approach a support system has been proposed to diagnose patients during trauma recovery using standards like WebRTC (Web Real-Time Communication) [11] and open source libraries like JSARToolKit (JavaScript Augmented Reality Toolkit) [12], ThreeJS [13] and NanoDICOM [14]. The idea behind this application was to use the patients DICOM images to create a 3D model that can be superimposed over the studied anatomical area of the body and use

WebRTC to share the augmented image with a remote web browser. By virtually attaching the 3D model to the real-time image of the patient, the model can rotate the same way as, for example, the subject rotates his leg, providing a digitally refined view to the physician. The main advantage during diagnosis or patient monitoring is that other colleagues can see the same image from a browser in another place without installing additional software components on the local operating system. This leads to a personalized diagnosis and/or recovery.

The second step was to use the results of the theoretical module adapting the system to gesture tracking devices and to visualize the rendered 3D model on a pseudo-holographic display providing a *natural user interface*, a term introduced by Steve Mann in his book on intelligent image processing [15], further refined by others as the technology for touch and mid-air gesture tracking devices evolved [16]. The Leap Motion controller is used for interacting with the application by hand gestures and a custom-made autostereoscopic display for visualizing the pseudo-holographic images.

In a third option, to show the generalization possibilities of the proposed models and solutions these were used for building a virtual laboratory for biomedical students. The system was adapted to a browser-based 3D molecular viewer. In this case GLmol [17] (a molecular viewer written in JavaScript) was extended by adding gesture tracking code using LeapJS [18] and ThreeLeapControls [19] as an interface between the two modules that allow the interaction with the rendered 3D molecular data.

The solutions presented in the thesis can be adapted to other areas where interacting with a rendered 3D model by multiple users without extra calibration steps is required. The web oriented nature and the open source modules of the developed systems assure their independence towards any specific operating system or proprietary browser extensions that otherwise could limit some important functionality, paving the way to port the applications to mobile devices.

## 1.2. Objectives of the PhD thesis

The main objective of the thesis is to consolidate theoretically and practically the field of the applications for medical image analysis using the latest generation of gesture tracking devices and autostereoscopic displays in a modular and personalized manner for specific users.

A secondary objective is to eliminate the need of having a specific operating system preinstalled and facilitate the development of mobile versions. The thesis proposes a new algorithm for image segmentation and colorization techniques adapted to a browser-based environment to cover this objective.

Modeling and development of educational virtual laboratories for the medical students is another objective.

The challenges faced at the technological level results from the novelty of both software and hardware instruments used for development and defining new metrics and parameters that will sustain the proposed solutions. In order to achieve these tasks the following specific activities were defined:

- a. A comprehensive analysis of the current software and hardware technologies that for 3D pseudo-holographic imaging.
- b. Development of a new segmentation algorithm that is fast enough to be applied in a browser-based medical visualization system.

- c. Implementing an interactive client-side colorization algorithm for the purpose of isolating the studied anatomical details by highlighting the regions of interest in the rendered medical images.
- d. Modelling and extending the functionalities of a web oriented 3D molecular viewer suitable for biomedical research, by adding gesture tracking for enhanced user experience.
- e. Evaluation of the developed software tools on target users using publicly available real medical image sets and 3D molecular models in order to validate the results based on specific test cases.

### 1.3. Structure of the thesis

The thesis consists of the following chapters that describe the technological state-of-the-art, the methods used to develop the applications, the results and the conclusions with further development opportunities:

- Chapter 2 is a technological overview of the new generation of input devices, brain-computer interfaces and pseudo-holographic display technologies that are suitable to build the next generation of natural user interfaces.
- Chapter 3 describes the open source software modules used during the research with emphasis on their role in the development process from JavaScript based Augmented Reality towards an open source client-side DICOM viewer that allows gesture-based interaction in a web browser.
- Chapter 4 presents the architecture, advantages, the problems and the solutions for the development of a web-oriented DICOM viewer based on the software technologies described in chapter 3.
- Chapter 5 works with open source libraries, modelling and developing extended functionalities for of a 3D molecular viewer adding gesture-based control of the rendered chemical models.
- Chapter 6 presents the evaluation procedure and results of the applications described in chapter 4 and 5.
- Chapter 7 presents the conclusions of the thesis and describes future directions for the research community.

Figure 1.1 represents the general overview and links between the chapters of the thesis.

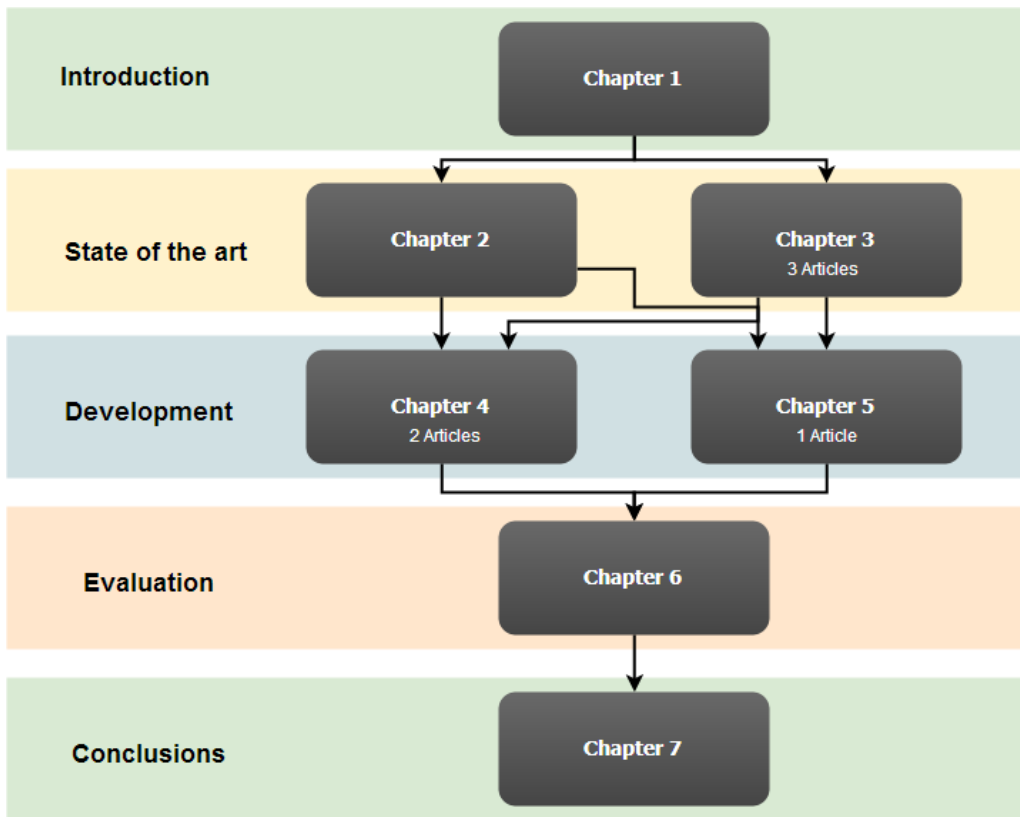


Figure 1.1. Structure of the thesis



## 2. HARDWARE TECHNOLOGIES OVERVIEW

This chapter is an overview of the latest hardware technologies involved in the development of state-of-the-art user interfaces. Any modern system architecture where multimodal interaction enhances the user experience may use the presented solutions as alternative input and output actions. There are also some uncommon BCIs (Brain-Computer Interfaces) mentioned that can enhance the life of patients in cases where they can use only the electrical signals from the brain to accomplish different tasks. The state-of-the-art hardware modules presented in the following sections are grouped in three categories: input devices based on gesture recognition, BCIs and some of the novel display technologies. Some of them are used to build the applications presented in Chapters 4 and 5 while others are presented here in order to understand the possibilities offered to further extend them.

### 2.1. Input devices based on gesture recognition

Gesture recognition represents a novel topic in computer science that allows the detection and tracking of various human gestures based on specific algorithms. In HCI this technique allows the user to point a finger towards an area on the 2D screen and associate the movement of the cursor with the movement of the hand, as we will see in some of the applications described in the thesis. After [20] when interacting with the content two separate types of gestures can be differentiated: *online gestures*, which can also be regarded as direct manipulations (like scaling, panning and rotating) and *offline gestures* that are usually processed after the interaction is finished (e.g. when a circle is drawn to activate a context menu).

The use of gesture recognition in HMI (Human-Machine Interaction) assures a natural way of interaction with the displayed 3D objects [21]. As shown in [5] the users can rapidly learn how to interact with a touch-less interface and to manipulate the displayed images even if they are holding other objects in their hands during the investigation of a patient.

Most of the latest gesture-based user interfaces [6, 22 and 23] are using Leap Motion or Kinect for detecting the users hand gestures or to track specific objects across video frames [24] by using the TLD (Tracking Learning Detection) algorithm [25]. These optical 3D sensors operating principles are based on different methods for scene analysis such as Structured Light, TOF (Time of Flight) or Stereo Vision. An analysis of these controllers operating principles can be found in [26].

Like with any other technology, developers of gesture-based user interfaces have to consider the eventual side-effects that might appear during exploitation. Such an undesirable effect was discovered in the 1980s when developers of vertically oriented touch-screens faced a problem that later become known as the "gorilla arm" [27]. The name comes from the fact that the users after interacting with a vertical display for a prolonged time, holding they arm horizontally and

making small motions, are feeling their muscles aching, cramped and heavy, like a gorilla arm. Later a quantitative method to assess arm fatigue during mid-air interactions is developed by a group of researchers that helps mitigate the problem called CE (Consumed Endurance) [28]. They are also proposing a series of guidelines for designing gesture-based interfaces that can avoid arm fatigue.

### 2.1.1. Leap Motion

The Leap Motion Controller (Figure 2.1) is a new generation of gesture tracking device that uses a pair of infrared cameras that can detect the motion of individual arms, hands and fingers with sub-millimeter accuracy at 200 frames per second and a 150 degree angle from its center. These features allow the development of low-latency applications even in a browser-based environment where usually the images are displayed at 60 frames per second.



Figure 2.1. Leap Motion interaction space

(source: [https://dv2ab2hvr9d0l.cloudfront.net/assets/product/how\\_1400-6f6bee27575c82645f61a2f590d19982.jpg](https://dv2ab2hvr9d0l.cloudfront.net/assets/product/how_1400-6f6bee27575c82645f61a2f590d19982.jpg))

Another key aspect to be considered by developers is that the Leap Motion Controller doesn't require any prior camera calibration, making it very suitable for collaborative applications where users can be changed instantly without having to wait for each other to finish the camera training.

From the hardware point of view the device consists of a pair of cameras that track infrared light with a wavelength of 850 nanometers [29] and three infrared LEDs (Light-emitting diodes) for illuminating the surrounding area (Figure 2.2).



Figure 2.2. Leap Motion hardware

(source: <https://cdn.sparkfun.com/assets/1/6/e/9/0/51c471f1ce395f0d73000001.jpg>)

The intensity of the light emitted by the three LEDs is limited by the maximum current that can be drawn over the USB (Universal Serial Bus) port used to connect the device to the computer. Due to this fact the viewing range of the sensor is limited to approximately 80 cm above the device making it impossible to accurately detect a larger area of the body other than the user's arms, hands and fingers.

The data read by the sensors takes the form of a stereo grayscale image pair as seen in Figure 2.3 used to extract the position of the objects seen by the infrared cameras.



Figure 2.3. Objects illuminated by the infrared LEDs  
(source: <http://blog.leapmotion.com/wp-content/uploads/2014/08/imageapi-hero.jpg>)

Even if the Leap Motion Controller should detect only the objects illuminated by its infrared LEDs, ambient environmental lighting sources such as light bulbs or the sunlight also emit light in the infrared spectrum affecting the behavior of the sensor.

Due to its low cost, robustness, high precision and low latency the Leap Motion Controller is a recommended gesture tracking device for applications where detecting only the users hands provides enough data in order to manipulate the virtually associated 3D objects.

### **2.1.2. MYO Armband**

MYO is a gesture control armband developed by Thalmic Labs that detects and interprets the electrical activity in the user's muscles (Figure 2.4) that is transformed into signals used to interact with various external devices such as computers, smart phones, toys, drones, etc. The device is equipped with medical grade stainless steel EMG (Electromyography) sensors, highly sensitive nine-axis IMU (Inertial Measurement Unit) containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer, according to the specifications found on the product site [30].



Figure 2.4. MYO detecting electrical activity in muscles  
(source: [https://static.thalnic.com/img/arm\\_with\\_signal.jpg](https://static.thalnic.com/img/arm_with_signal.jpg))

The MYO armband even if it needs a prior calibration step, after it learns to distinguish between the different signals produced by specific muscles can transform these patterns to command external devices via a wireless Bluetooth 4.0 connection.

In a recent study [31] the MYO armband was successfully used to command an MPL (modular prosthetic limb) as shown in Figure 2.5.



Figure 2.5. Two MYO armbands used to control a modular prosthetic limb  
(source: [http://www.jhuapl.edu/newscenter/pressreleases/2016/images/20160112\\_8267\\_lg.jpg](http://www.jhuapl.edu/newscenter/pressreleases/2016/images/20160112_8267_lg.jpg))

The researchers from the Johns Hopkins University Applied Physics Laboratory and from the Division of Musculoskeletal Oncology at the University of Pittsburgh Medical Center, successfully used osseointegration to attach a prosthetic arm directly to the residual limb of an amputee, Johnny Matheny, who lost his left arm in 2008 due to cancer. The patient was also subject to a surgical procedure called TMR (targeted muscle reinnervation) that rewires the connections of the nerves that once were used to control the missing limb, enabling the control of a motorized prosthetic device and in some cases to regain sensory feedback.

### 2.1.3. SideSwipe

In a recent paper [32] a team of researchers from the University of Washington are presenting a system called SideSwipe (Figure 2.6) that uses the actual GSM (Global System for Mobile Communications) signal to detect in-air hand gestures both above and around a mobile device.

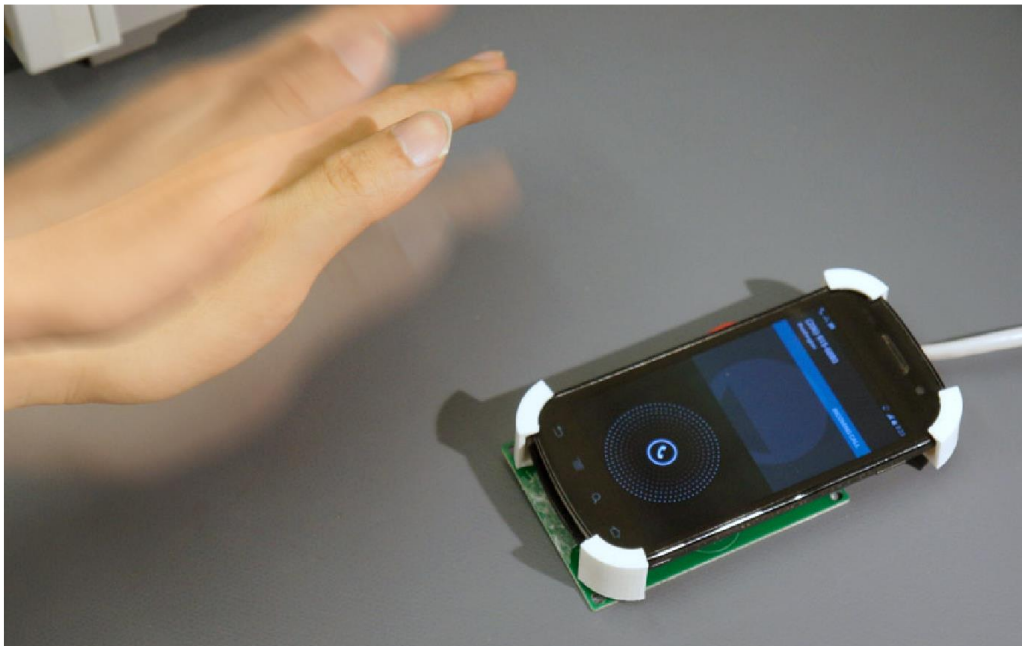


Figure 2.6. The SideSwipe system after [32]

The system's working principle is based on the fact that when the user moves his hand in the proximity of the mobile phone his gestures are affecting the signal propagation between the device's transmitter and the receiving antenna array found on the PCB (Printed Circuit Board) of the SideSwipe system.

SideSwipe is capable to distinguish between three categories of gesture sets such as tap, hover and slide. These gestures can be combined to form a more complex vocabulary and assigned to different tasks that the applications of the phone may accomplish.

#### 2.1.4. SoundWave

SoundWave [33] is a sound-based gesture detection technique that uses the speaker and microphone of a computer system to detect in-air gestures (Figure 2.7).



Figure 2.7. SoundWave in-air gesture sensing after [33]

The technique is based on the Doppler effect (also known as Doppler shift) a phenomenon which characterizes the change in frequency of a sound wave as an observer moves closer or farther away relative to its source. Based on this effect, SoundWave uses the existing speakers of a device in order to generate tones between 18-22 kHz, which are for most people out of the audible range. After that the reflected sounds are picked up by the microphone and by analyzing the frequency shifts the system is capable to distinguish the speed, direction and the amplitude of the motions in the proximity of the computing device. This large variety of the detected parameters can be assigned to accomplish various tasks during the interaction with the device. For example, as described in [33], “the direction and speed of a hand moving up or down can be sensed to scroll a webpage in real-time” or by detecting two hands that are moving in opposite directions can

be used as a rotation gesture making it very similar to the actual gesture used to rotate a real object with two hands.

The most important aspect is that SoundWave can detect gestures without LoS (line of sight) and only uses the built-in speakers and microphone facilitating the use of the technique on mobile devices as later demonstrated [34].

The AirLink [35] is a system based on the same principle as SoundWave and allows the interaction between multiple devices using in-air gestures as shown in Figure 2.8.

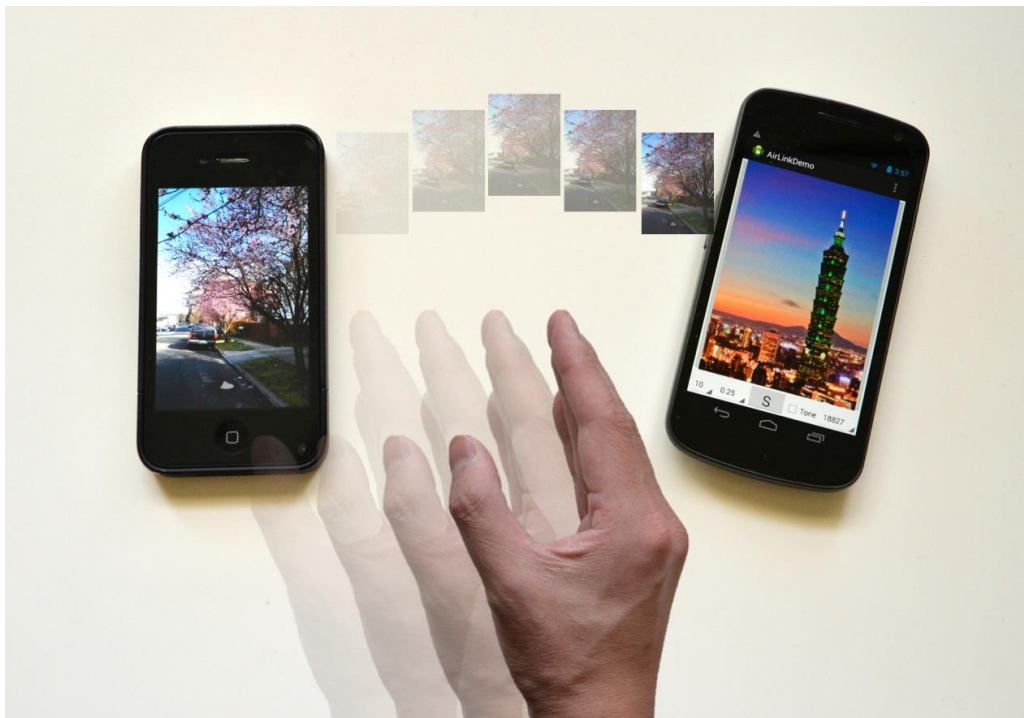


Figure 2.8. AirLink interaction between multiple devices after [35]

By using this novel technique, users can directly transfer files between multiple devices such as mobile phones, tablets or laptops simply waving their hand from one device towards another. The system is capable to establish the relative positions of the phones when a swipe gesture is performed or to share files from one to multiple devices by using a central server to detect the initiating device.

### 2.1.5. Project Soli

Soli [36] is a novel gesture sensing technology, developed by Google's Advanced Technologies and Projects division that come in a form of a very small (5x5 mm) low-power radar sensor which can translate the users hand gestures into digital signals (Figure 2.9).

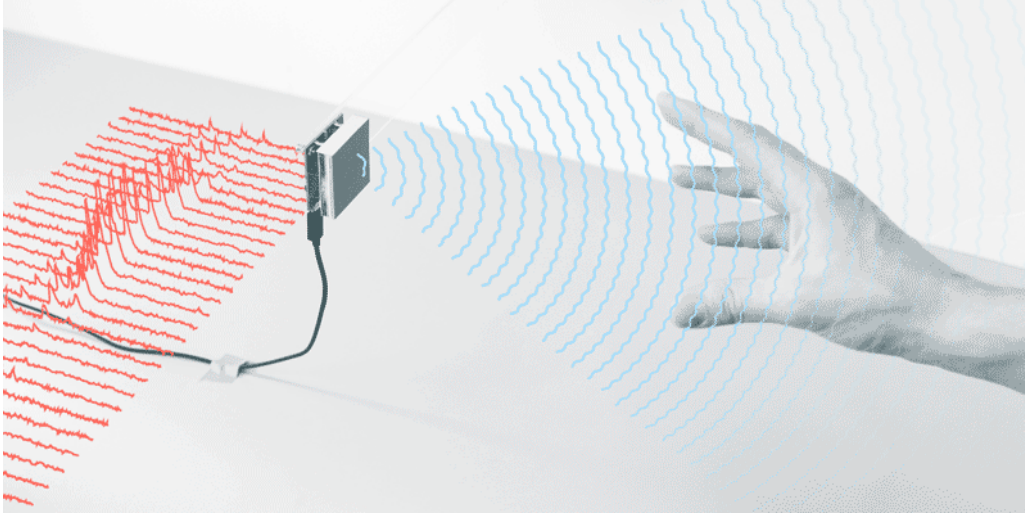


Figure 2.9. Soli converting gestures into digital signals  
(source: <https://atap.google.com/soli/static/img/what-is-soli/radar-visualisation-loop.gif>)

According to the project page the radar operates in the 60 GHz ISM (industrial, scientific and medical) radio band and comes with two modulation architectures: a FMCW (Frequency Modulated Continuous Wave) radar and a DSSS (Direct-Sequence Spread Spectrum) radar.

The small size of the chip (Figure 2.10) makes it a perfect candidate for inclusion into future wearable devices.

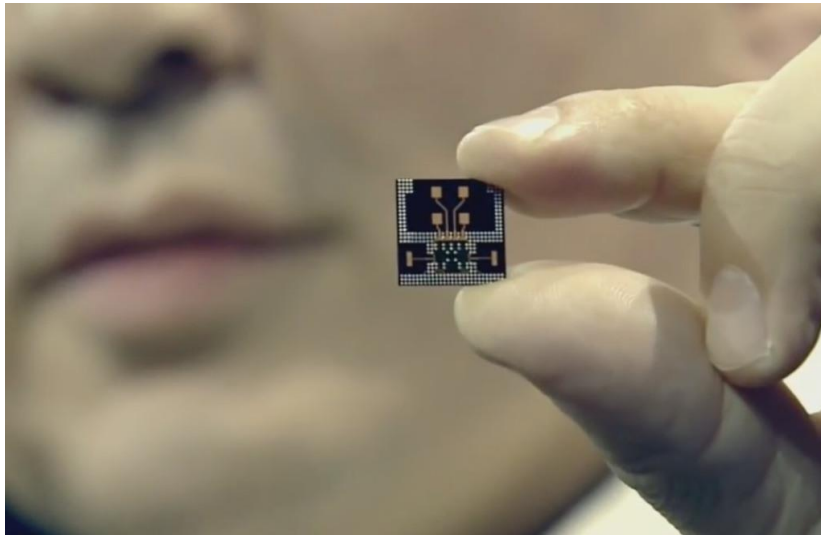


Figure 2.10. Soli chip and antennas  
(source: <https://tctechcrunch2011.files.wordpress.com/2016/05/soli.jpg?w=1024&h=658>)



The major advantage of the system is that it operates at a very high frame rate from 100 to 10000 frames per second, much faster than any other actual motion detection system.

The project team has already built a working prototype and they are testing the radar with smart watches, the current concern being only to reduce the power consumption of the device and to launch the development kit as soon as possible in order to exploit the technology at its maximum potential.

## **2.2. Brain-computer interfaces**

BCIs are systems that allow the direct communication between the CNS (Central Nervous System) and a computer. For this reason they are also often referred as DNIs (Direct Neural Interfaces) and used interchangeably with the term neuroprosthetics that usually connect the peripheral nervous system to an external device (e.g. a prosthetic arm, leg or cochlear implant).

Even if the researches in the BCI area begin in the 1970s [37], this type of connections would not be possible without Hans Berger's study of the electrical activity of the brain and the development of a monitoring method of this activity called EEG (electroencephalography).

Based on the fact that surgery is used or not in order to make the connection between the sensors and the nervous system, BCIs fall in to one of the following categories:

- Non-invasive BCIs which do not require any type of surgery in order to acquire electrical signals from the brain and the electrodes are usually placed on the patients scalp like in EEG. Functional neuroimaging techniques, such as fMRI (Functional Magnetic Resonance Imaging) or MEG (Magnetoencephalography), fall in the same category.
- Invasive BCIs, like ECoG (Electrocorticography) are implanted using neurosurgery and they provide the best results regarding the reading of the electrical signal from the brain that otherwise can be reduced by the skull bone leading to inaccurate results.

The most impressive use of BCIs is detecting awareness in patients with DOC (Disorders of Consciousness) [38]. This state includes patients who are in coma, VS (Vegetative State) or MCS (Minimally Conscious State), but BCIs may be a valuable tool also for patients who present a LIS (Locked-In Syndrome) [39]. These patients are unable to communicate their needs or consent regarding treatment decisions towards the physicians. In most cases their families are put in very sad situations when they have to make decisions regarding the end-of-life of their loved ones. With the help of BCIs this can become the decision of the patient himself, as it is with any other case when euthanasia might come into discussion.

### **2.2.1. EPOC Plus headset**

The EPOC+ headset [40] produced by Emotiv is a non-invasive neural interface (Figure 2.11) based on reading and interpreting of the users EEG signals and used in neurogaming (interacting with a computer game without a conventional controller).



Figure 2.11. EPOC headset

(source: [http://emotiv.com/wp-content/uploads/2016/02/emotiv\\_epoc\\_600.png](http://emotiv.com/wp-content/uploads/2016/02/emotiv_epoc_600.png))

The headset has 14 EEG sensors and 2 references for reading the electrical signals produced by the user's brain, a nine-axis IMU (like the MYO armband) and is connected to the computer wirelessly via Bluetooth SMART 4.0 LE.

The EPOC comes with several detection suites that allow it to distinguish between:

- facial expressions like blink, left wink, right wink, furrow (frown), raise brow (surprise), smile, clench teeth (grimace), glance left, glance right, laugh, smirk (left side), smirk (right side);
- emotional states such as instantaneous excitement, long term excitement, frustration, engagement, meditation, interest / affinity;
- mental commands, to manipulate virtual objects, like push, pull, lift, drop, left, right, rotate clockwise, rotate anti-clockwise, rotate forwards, rotate backwards, rotate left, rotate right, disappear;

The headset doesn't allow direct access to raw EEG signals, but this can be purchased separately and in this case the EPOC will be equipped with a special firmware that will allow the display of the signals by their TestBench software.

Even if the spatial resolution of EEG systems is less accurate than the fMRI devices (where it can be 1mm), they compensate by the speed of reading the electrical signals that can be at the magnitude of milliseconds sometimes.

EEG based neuroheadsets have multiple advantages such as low production costs, wireless connection, tolerance towards patients head movements, they are quiet during operation and most important that they do not require invasive surgery to be installed (e.g. EPOC uses saline based wet sensors).

### 2.2.2. Functional Electrical Stimulation

FES (Functional Electrical Stimulation) is a technique of applying controlled electrical impulses to the muscles of individuals who suffered a SCI (Spinal Cord Injury). These impulses applied in a specific pattern are inducing a series of contractions and relaxations in the muscles generating a functional movement of the hands [41] or legs [42] based on the patients situation. Depending on the application of FES, the electrodes used to transmit the electrical impulses can be applied on the skin, above the paralyzed muscles, or surgically inserted directly on the nerves that communicate with the muscles (Figure 2.12).

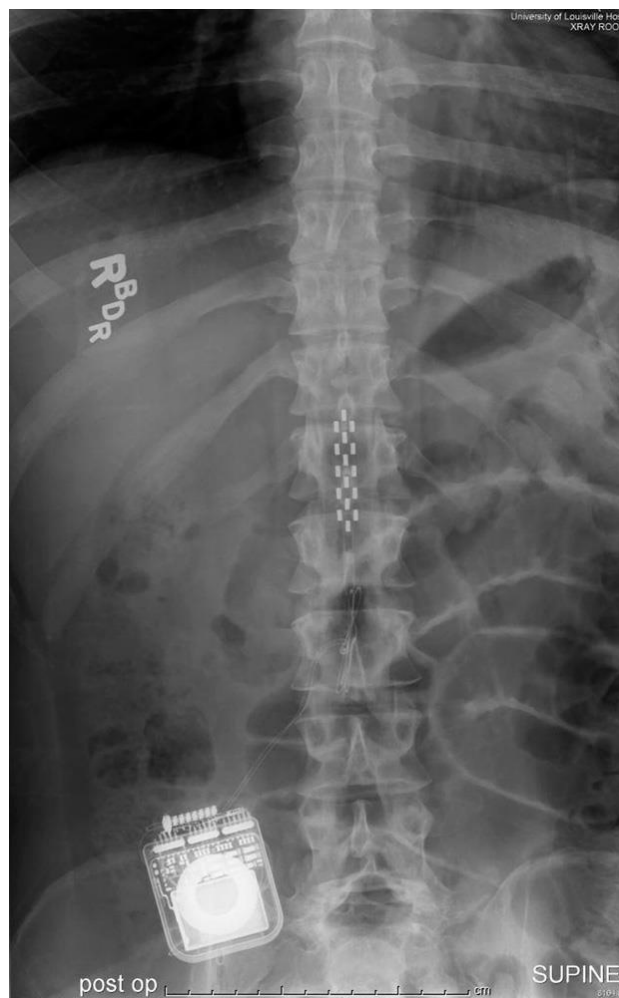


Figure 2.12. Electrode array implanted next to the spinal cord and a pulse generator below (source: <http://spectrum.ieee.org/img/110LSpinalStimulationXRy-1381869862172.jpg>)

One of the applications of the technique is FES cycling (Figure 2.13), an activity that helps patients preventing obesity, cardiovascular disease, diabetes and to improve their condition by decreasing the risk of fractures by reversing bone loss [43]. Regular exercise can elevate patients mood and helps preventing depression, a condition often faced by patients who excessively rely on wheelchairs due to SCI after accidents.



Figure 2.13. Patient performing FES cycling  
(source: <http://www.medphys.ucl.ac.uk/research/impdev/fits/Images/esplanade8a.gif>)

A recent article [44] describes the results of neuroscientist Susan Harkema from the University of Louisville, Kentucky Spinal Cord Injury Research Center, who managed to induce conscious control over voluntary movement, standing, and assisted stepping to a patient with a complete loss of clinically detectable voluntary motor function using epidural stimulation of the lumbosacral spinal cord [45]. The novelty of this approach is that the patient was able to fully support his weight in standing position (without a harness), voluntarily move his toes, ankles and flex his legs during epidural stimulation. The subject also regained partial control of his bladder, sexual function and temperature regulation, even after the electrodes were turned off [44].

### 2.2.3. Osseointegrated human-machine gateway

The researchers at the Chalmers University of Technology developed a first and unique neuroprosthetics called OHMG (osseointegrated human-machine gateway) [46] using an OPRA (Osseointegrated Prostheses for the Rehabilitation of Amputees) implant system [47]. The OPRA implant system assures a stable mechanical fixation required for load transfer from the prostheses towards the bone. With a little customization of the abutment-screw and the central sealing component (normally these components do not need to transmit electrical signals to a regular

prostheses) the implant provides an electrically stable skin interface necessary to transmit the signals between the peripheral nervous system and the prosthetic limb (Figure 2.14).

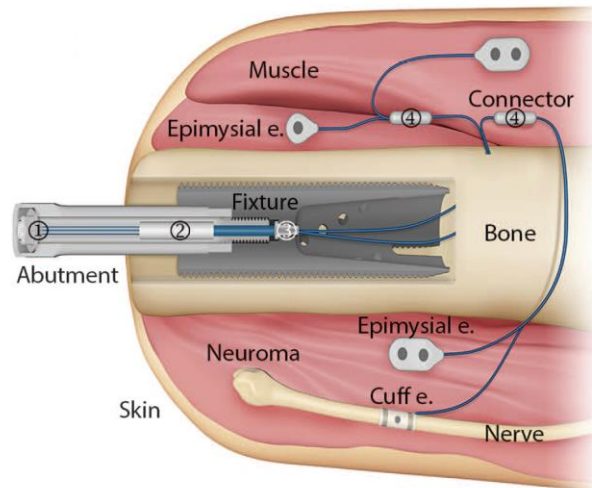


Figure 2.14. The connection of the OHMG system to the peripheral nerves after [46]

The novelty of the OHMG is that allows bidirectional communication between the attached prosthetic limb and the surgically implanted electrodes in the muscles, the patient being able to have a tactile perception at some extent [46]. The OHMG improves the degree of control over the prosthetic arm in comparison with other systems where surface electrodes are used [31] as a result of this the patient is able to perform his daily living and professional activities (Figure 2.15).

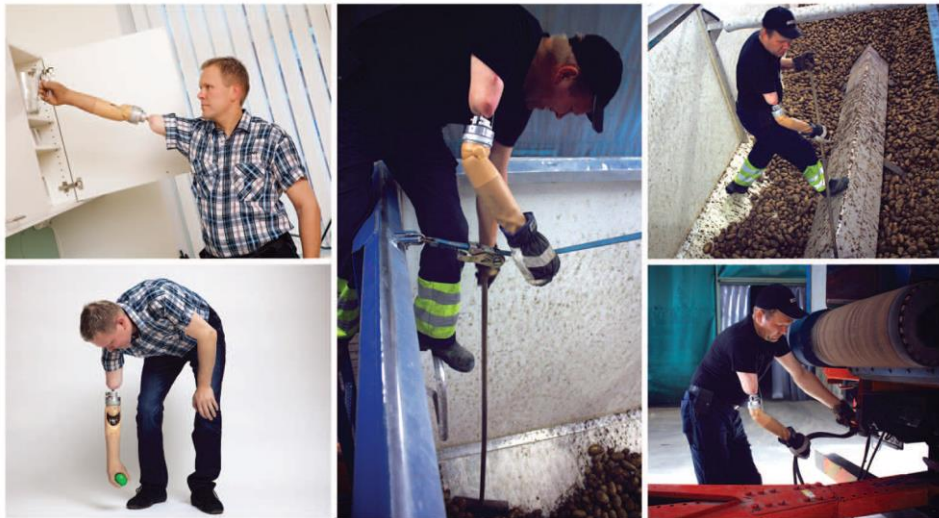


Figure 2.15. Patient performing his daily activities with the OHMG system after [46]

The researchers are continuing the development and the testing of the OHMG since only one patient benefitted so far from the results. As shown in [46] the stable attachment and thought based control of an artificial prosthesis is a clinically viable solution that assures an increased daily activity level and better quality of life for patients who lost their limbs.

#### 2.2.4. BrainGate

BrainGate [48] is a DNI that can be surgically implanted to patients that suffer from severe forms of paralysis following SCI, LIS, ALS (Amyotrophic Lateral Sclerosis) or brainstem stroke, which hinders the direct communication between the CNS and the rest of the body. The patients are able to control of a robotic arm to perform reach and grasp movements (Figure 2.16), without explicit training, using the signals provided by a 4x4 mm, 96-channel microelectrode array implanted in the motor cortex.

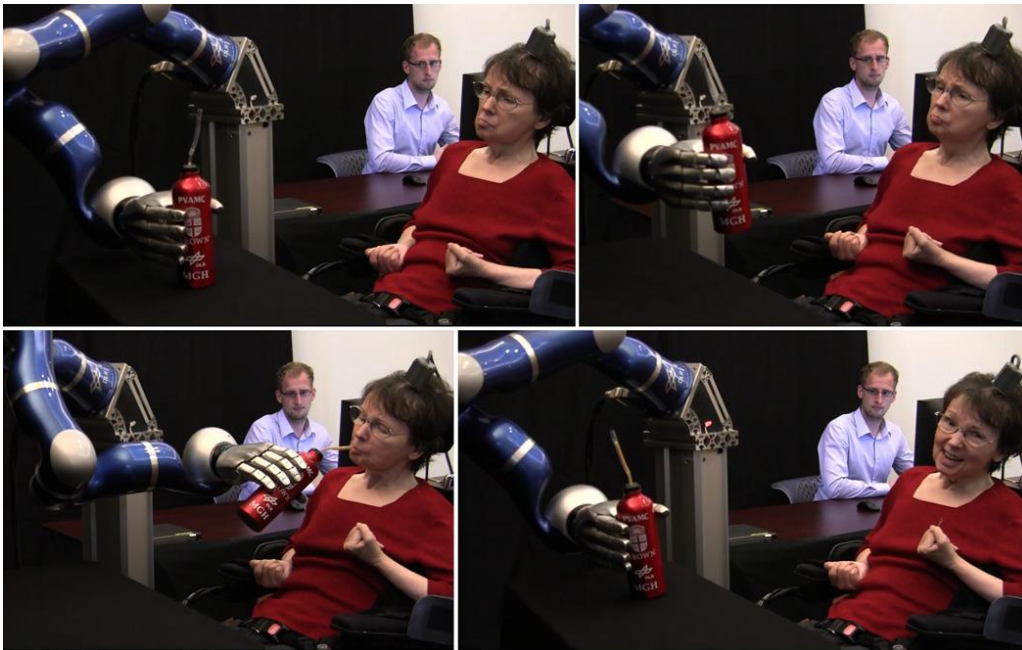


Figure 2.16. Patient drinking from a bottle using BrainGate after [48]

The main advantage of such an implant is that a patient can interact with the surrounding environment much independently and with a personal computer by onscreen keyboards or moving the cursor on the screen. The disadvantage comes from the postoperative risks imposed by any medical intervention at such a level. In order to avoid such problems other HCIs are more likely to be recommended for patients with severe paralysis, such as ones based on pupillometry [49].

### 2.3. 3D Display technologies

The classic 2D display technologies, like CRTs (Cathode Ray Tubes), plasma display panels or LCDs (liquid crystal displays), due to their construction principle, are meant to present only a 2D view of the displayed images even if the subject very often might be a higher-dimensional 3D dataset (e.g. 3D movies, 3D games, 3D medical image datasets). These technological limitations are interfering with the human brains ability to properly distinguish the depth related visual information and very often lead to misinterpretation of the displayed 3D images. These distorted and unnatural spatial representations of real-world 3D objects are very successfully used in various art forms to create optical illusions, but can be deceptive to professionals who are seeking for a better understanding of a particular 3D subject.

Current 3D computer graphics are generated using mathematical algorithms that can emulate at some extent the visual depth cues needed for a proper interpretation of the displayed images. The CGI (computer-generated imagery) displayed on a traditional 2D monitor even if provides a more accurate representation of a real 3D object, are still lacking some important aspects that can be reproduced only by a 3D device. In order to understand these aspects we need to understand the visual depth cues used by the human brain in producing the 3D depth sensation based on what we see.

According to [50] and [51] the human visual system needs both physical and psychological depth cues to recognize the third dimension and to produce a proper depth sensation. Although there are several other depth cues mentioned in the literature, according to the author of [50] we can distinguish four major physical (physiological) and five psychological ones used by the brain in order to gain a true 3D sensation. The physical depth cues used by the human visual system are:

- *Accommodation* – the measurement of the tension of the muscle that is used to force the eyes' lenses to change their focal length and bring into focus a specific real-life object in the viewed 3D scene, resulting in the proper perception of its 3D depth.
- *Convergence* – the angular difference between the viewing directions of each of the viewers' eyes when looking at the same spot on a 3D object at the same time. The closer the subject is located to the viewer, the more the eyes are converging. In some medical conditions (e.g. strabismus, convergence insufficiency) due to the misalignment of the two eyes this can lead to double vision (diplopia). The direct effects of such insufficiencies are: eyestrain, headaches, blurred vision, inability to concentrate, sleepiness, motion sickness, vertigo etc..
- *Motion parallax* – offers depth cues by comparing the relative motion of the objects viewed in a 3D scene. This causes nearby objects to appear to move faster than those that are farther away when the viewer moves his head.
- *Binocular disparity* – also referred to as binocular parallax is based on the fact that the viewers' eyes are seeing two, slightly different perspectives of an object in a 3D scene. The farther away from each other the perceived images appear to be, the farther away the real-world 3D object will be.

Due to the adaptive nature of the human brain, in its constant attempt to make sense of the surrounding world, the human visual system is capable of perceiving the depth sensation by extracting psychological cues from plain 2D monocular images. After [50] these psychological depth cues are:

- *Linear perspective* – the illusion that two parallel lines converge at a distant vanishing point on the horizon. This technique is used in art since the Renaissance in order to create a more realistic representation of various structures such as buildings, anatomical body parts and natural landscapes, creating the impression that the paintings have a 3D depth and they are depicting a 3D volume.
- *Occlusion* – also referred to as *overlapping*, is the ability of the human brain to perceive partially occluded (overlapped) objects in a 3D scene as being farther away than the ones in front of them.
- *Shading* – inside a 3D scene helps the human visual system to interpret the spatial orientation and the surface shape of the viewed 3D objects, based on the orientation and the intensity of the shadows cast by one object upon another.
- *Texture* – provides an important visual cue due to the fact that usually objects with smooth textures are interpreted as lacking some of their surface details, tricking the brain to perceive them as being farther away than their real position in a 3D scene.
- *Prior knowledge* – is one of the most important cues used by the human brain to constantly analyze and compare the viewed 3D shapes to the familiar 3D objects that were previously seen, because we already know how a specific geometrical structure will interact with the surrounding light when it moves in front of us. This visual cue leads to a faster processing of the viewed scene and since it is not always accurate it can be successfully exploited in magic shows. The term is referred to by illusionists and psychologists as cognitive blindspot or cognitive bias.

Based on their construction techniques, some of the current 3D display devices are capable to provide a wide range of physiological and psychological depth cues, leading to an enhanced visual experience, while others might provide conflicting depth cues that could lead to eye fatigue and nausea. In the case of volumetric displays the shading might be misleading due to the fact that the displayed voxels are transparent, providing a ghostly appearance. In the worst case some of the stereoscopic 3D displays (e.g. in the case of 3D movies where special glasses are used to provide the depth sensation), based on the 3D content, the accommodation and convergence information might be conflicting because the displayed images are closer (on the movie screen), not at their perceived distance in space. This disparity often leads to motion sickness due to the fact that the brain reacts by adjusting the viewers' body position trying to synchronize the information perceived by the vestibular system with the visual cues provided by the eyes. The phenomenon is often referred to in literature as accommodation/convergence breakdown or A/C breakdown [52].

Modern 3D display technologies can be divided into two major categories:

- *stereoscopic displays* – when special glasses or headgears are used to provide the 3D depth sensation. This category includes color-interlaced (anaglyph), polarization-interlaced, time-multiplexed (relies on persistence of vision) and head-mounted displays, and they mostly provide two slightly different perspectives of the same image (a stereo



image pair) for each eye that are merged together by the human visual system, thus providing the illusion of 3D depth;

- *autostereoscopic displays* – also referred as *glasses-free 3D displays*, where various techniques are used to induce the binocular 3D depth sensation without the use of any kind of special glasses. According to [50] we can further divide this category into multiview 3D displays, volumetric 3D displays and holographic displays, despite the fact that the latest are often also referred to as pseudo-holographic displays because they are not using actual holograms.

Due to the constant changes and improvements regarding the development of the current 3D display technologies only the major display types will be discussed further in order to present an overall view of the present state of the art.

### 2.3.1. Stereoscopic displays

Stereoscopic displays (Figure 2.17), also known as 3D displays, are constrained by the use of special glasses in order to provide the 3D depth perception (the stereopsis).



Figure 2.17. Clarity Matrix 3D display by Planar

(source:

[https://vrlogic.com//images/Stereo\\_Displays/PLANAR/Clarity/PLANAR\\_Clarify\\_Header.png](https://vrlogic.com//images/Stereo_Displays/PLANAR/Clarity/PLANAR_Clarify_Header.png))

The 3D sensation in all cases is achieved by providing an image pair (like a stereogram), that represents two different perspectives of the viewed subject, that are merged together inside the human brain's visual cortex, resulting in an illusion of stereoscopic depth of the viewed 3D scene. There are several techniques used to separate the two images presented to the left and right eye of the viewer and usually they imply spectral (color-interlaced), polarization (polarization-interlaced) and temporal (time-multiplexed) differences between the two images.

The major disadvantage of this set of technologies used in the construction of stereo 3D displays is that they are prone to A/C breakdown as previously discussed.

### 2.3.2. Autostereoscopic displays

Autostereoscopic (multiview) displays do not require the use of any kind of special glasses in order to provide a perception of 3D depth. This type of 3D displays are usually built on techniques such as parallax barriers [53, 54] or lenticular lenses [55, 56] and are providing a finite number of views (different perspectives of the same 3D object) to a specific number of viewers as shown in Figure 2.18.

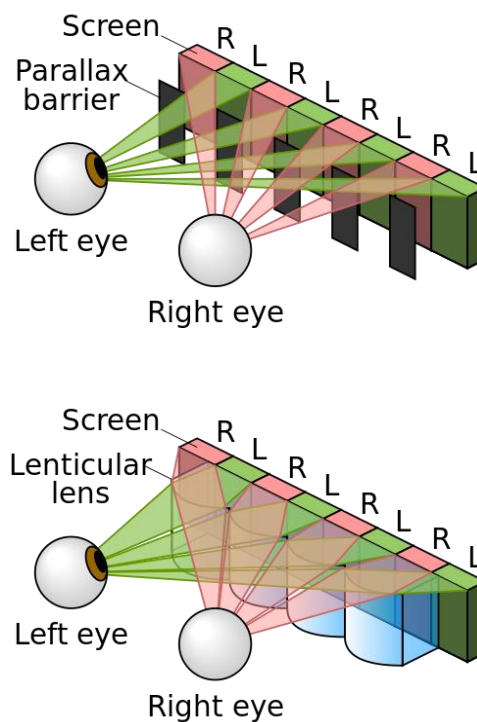


Figure 2.18. Operating principle of a parallax-barrier vs. a lenticular lens

(source:

[https://upload.wikimedia.org/wikipedia/commons/thumb/a/a8/Parallax\\_barrier\\_vs\\_lenticular\\_screen.svg/500px-Parallax\\_barrier\\_vs\\_lenticular\\_screen.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/a/a8/Parallax_barrier_vs_lenticular_screen.svg/500px-Parallax_barrier_vs_lenticular_screen.svg.png)

The parallax barrier technique has been successfully adapted by Land Rover in 2010, when the company provided a central display on their infotainment system that was capable to display two different images at the same time, one for the driver and one other for the passenger [57]. This way while the driver could get GPS directions, the passenger could watch a movie at the same time, both of them using the same display.

Since the displays based on lenticular lenses and parallax barriers are blocking a specific set of pixels for each viewer they provide a lower resolution. In order to surpass such shortcomings other techniques have been developed that allow multiple users to see the same or different perspectives of a 3D object without using any kind of glasses. Although these technologies might be also referred as autostereoscopic displays they must be treated separately from the main trend since they have little to none of the resolution limitations imposed by the regular autostereoscopic displays.

### 2.3.3. Volumetric 3D displays

Volumetric 3D displays are a special type of autostereoscopic devices that are capable of showing accurate 3D volumetric information of the rendered 3D models. These displays are using individual *voxels* (volume elements) that can be manipulated separately by the underlying software. The major advantage of the volumetric 3D displays is given by the fact that they provide a 360 degree view of the 3D model not only a limited number of separate perspectives as in case of the devices based on parallax barriers or lenticular lenses.

Based on their construction type the volumetric 3D displays can be divided into two major categories:

- *Static screen volumetric 3D displays* – that have static components that can be manipulated as individual voxels in order to form a 3D image as in Figure 2.19.

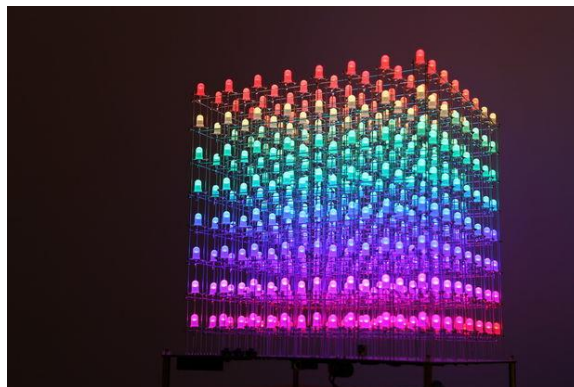


Figure 2.19. A static screen volumetric 3D display formed by RGB LEDs

(source: [http://forum.allaboutcircuits.com/proxy.php?image=http%3A%2F%2Fcdn.instructables.com%](http://forum.allaboutcircuits.com/proxy.php?image=http%3A%2F%2Fcdn.instructables.com%2F)

2FFKU%2FPRAQ%2FIA593H7Y%2FFKUPRAQIA593H7Y.MEDIUM.jpg&hash=93156183651da40314d089a45938157d)

- *Swept screen volumetric 3D displays* – that involves moving components such as a projection screen formed by a rotating mirror that reflects the light provided by a projector as in Figure 2.20. In this case the 3D image is formed by a series of individual depth planes (2D slices of the 3D image) that are projected on the mirror and are fused together by the human brain due to the persistence of vision phenomenon. The same principle is used in the classical CRT monitors, with the major difference that swept screen volumetric 3D displays are capable of displaying actual 3D datasets, not only individual 2D pixels.



Figure 2.20. Volumetric display using a moving projection screen  
(source: <http://i.stack.imgur.com/pO0M2.jpg>)

A Japanese company, called Burton Inc. [58], has a revolutionary approach in building swept screen volumetric displays that completely eliminates the classical

projection screen. Their technology involves focused lasers that heat the air near a well controlled 3D position in space (in the focal point) resulting in a plasma emission. By changing the position of the focal point of the laser in midair the Aerial Burton 3D display is able to form various patterns based on individual dots that are forming a 3D image as shown in Figure 2.21.

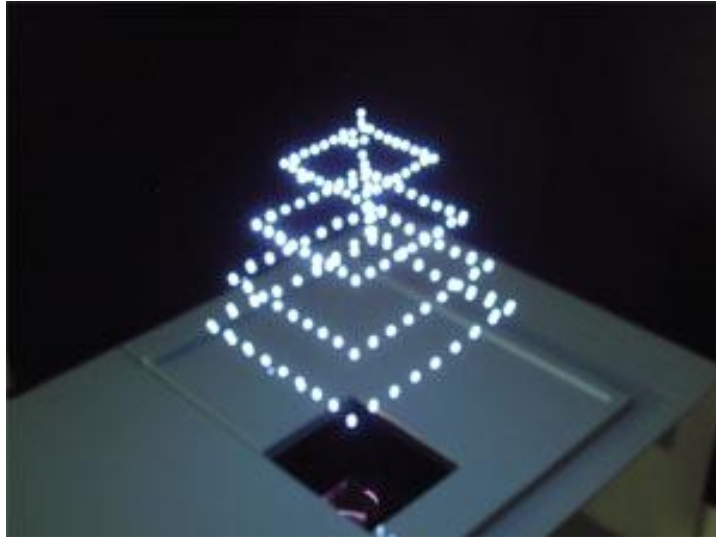


Figure 2.21. Aerial Burton 3D display  
(source: <http://cdn.phys.org/newman/csz/news/800/2006/2-fig1.jpg>)

The advantages of the volumetric 3D displays consist of the fact that they are providing the most information regarding the displayed 3D subject, since the number of views displayed at the same time is virtually unlimited. As a disadvantage the users cannot touch the displayed images without the risk of serious injuries due to the moving components like in case of the mirrors or burning themselves as in case of the laser-based solution of the Aerial Burton 3D display, however the skin damage might be reduced by using femtosecond laser systems as described in [59].

#### 2.3.4. Pseudo-holographic displays

Pseudo-holographic displays are based on displaying individual hogels (holographic picture elements) a term introduced by Mark Lucente in his doctoral thesis [60]. This type of autostereoscopic displays are providing an illusion of a 3D image that seems to float in midair that resembles to an actual hologram, despite the fact that the 3D model and the associated lights are computer generated.

There are several construction techniques used for current pseudo-holographic displays, but they are all based on the *Dircksian Phantasmagoria* projection technique developed by a British engineer Henry Dircks [61] in the 19<sup>th</sup> century. This was later popularized by John Henry Pepper who registered a patent for the technique in 1863 [61] and named it *Pepper's ghost*, even if he always recognized Dircks as the initial inventor. The *Pepper's ghost* technique implied the

use of an angled sheet of glass placed on the stage that would reflect an actor hidden beneath the stage when the room he was hiding in was properly lit by an assistant as shown in Figure 2.22. This way the hidden actor could perform alongside his colleagues even if he was not physically on the stage, resulting in a dramatically and ghostly appearance.

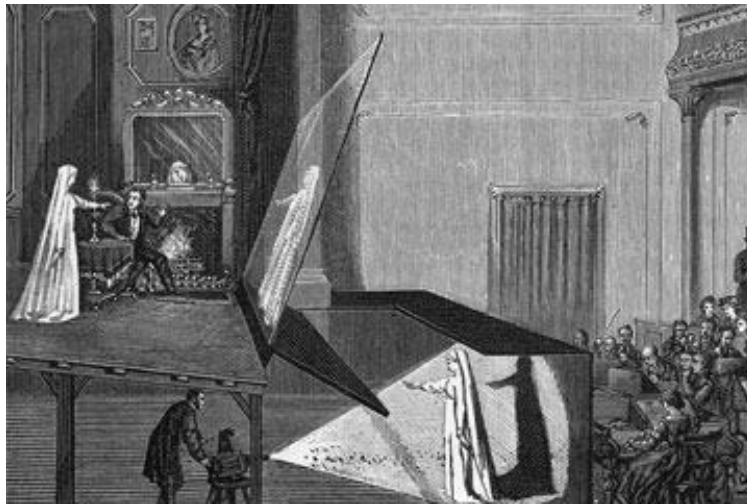


Figure 2.22. Pepper's ghost technique used in theaters  
(source: <http://s.hswstatic.com/gif/projections-celebs-2.jpg>)

Pseudo-holographic displays based on their construction techniques are capable of providing one or multiple perspectives of a 3D model to multiple users at the same time without the use of any kind of 3D glasses.

During the development phase of the applications, later discussed in the thesis, the author of the thesis used a HoloVision pseudo-holographic 3D display that was patented [62] and manufactured by the Provision Holding, Inc. [63], successfully used in stores for advertising various commercial products as shown in Figure 2.23.



Figure 2.23. VisPod 73D (HoloVision) pseudoholographic display by ProVision  
(source: Wonderworks Media Ltd. user manual)

The HoloVision autostereoscopic 3D display allows multiple users to visualize a floating 3D object without the need for special 3D glasses. The device creates a vivid illusion of a 30 cm image that is focused at approximately 90 cm in front of the 3D display and is viewable at an approximate 60-degree angle from the center up to 10 meters and even more. The technology used is very promising since it can be used for interactive lessons, where multiple students can see the rendered 3D model at the same time.

Inverted pyramid displays, such as the DREAMOC HD3 [64], are built in a pyramidal shape that allows the projection of multiple views of the same 3D object on each face by using a 2D monitor placed on top as shown in Figure 2.24.



Figure 2.24. DREAMOC HD3 pseudoholographic display by Realfiction  
(source: <http://www.dreamoc.nl/wp-content/uploads/2013/10/HD3.jpg>)

This technique implies the development of special 3D content, since the images projected on each face of the pyramid have to reflect the corresponding face of the displayed 3D model.

### 2.3.5. Holographic film projections

Holographic film projections are a special type of pseudo-holographic displays that allow the blending of the projected images with the surrounding reality the same way the original Pepper's ghost technique did. This technique implies the use of a special holographic film that usually contains a large number of lenticular lenses that focuses the light and enhances the resolution of the displayed images provided by a regular projector that can be mounted in front or the back of the film as shown in Figure 2.25. The film usually is mounted between two sheets of glass making it very suitable for marketing purposes.



Figure 2.25. Holographic film projection  
(source: <http://image.made-in-china.com/2f0j00TKetDkGcnZqm/HD-Holographic-Projection-Screen-with-Transparent-Fabric.jpg>)

The holographic film is produced with various transparency levels that can provide a very vivid image suitable for telepresence applications.

### 2.3.6. Heliodyisplay

Heliodyisplay [65] is an air-based display developed by IO2 Technology [66] that uses conventional air as a projection screen as shown in Figure 2.26.



Figure 2.26. Heliodyisplay  
(source: <http://www.blogcdn.com/www.engadget.com/media/2007/02/2-28-07-heliodyisplay.jpg>)



The major breakthrough is that the users can interact with the displayed 3D images using gestures like on a regular touch screen and that the device uses dry atomized air so it would not interfere with other electronic components due to humidity like in the case of vapor-based displays.

### 2.3.7. Braille terminals

A braille display is a HID (Human Interface Device) that is used to display braille characters for blind users who cannot use a regular display. The characters are formed by a series of rounded pins that are grouped and raised in patterns that are detected by the user with his finger tips as shown in Figure 2.27.

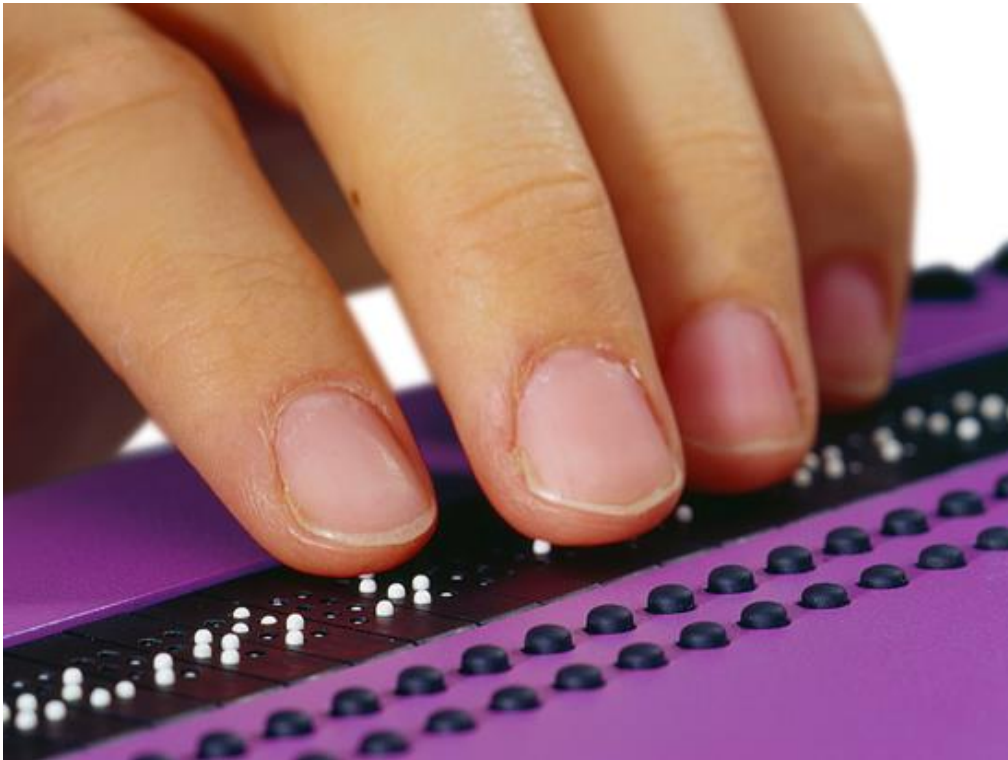


Figure 2.27. Braille terminal  
(source: [http://www.indes.eu/wp-content/uploads/2012/10/alva\\_slider2.jpg](http://www.indes.eu/wp-content/uploads/2012/10/alva_slider2.jpg))

In 2015 Tactisplay Corp. [67] went further with the idea and developed a graphic braille display that has a 1000 braille cells formed by 120 columns and 100 rows, that is capable to show more elaborate content such as BANA (Braille Authority of North America) compatible braille graphics as shown in Figure 2.28.

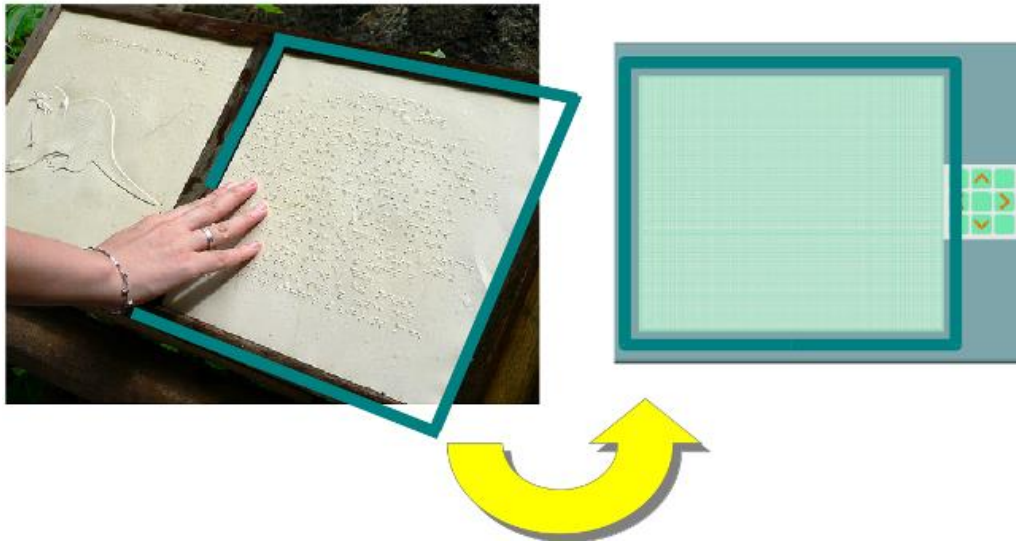


Figure 2.28. Tactisplay displaying graphic content

(source: [http://www.tactisplay.com/\\_/rsrc/1468860877623/product/tactisplay-table/braille%20graphics%20in%20one%20page-s3.png](http://www.tactisplay.com/_/rsrc/1468860877623/product/tactisplay-table/braille%20graphics%20in%20one%20page-s3.png))

Even if the screen update time is 8 seconds, this braille terminal can display complex equations, graphs, spreadsheets and shapes of simple images on the same page with text content enhancing the learning process for the blind or deafblind persons.

Although this technology is meant to display text content that is shown on a regular 2D monitor, it paves the way to the development of the haptic devices used to interact with 3D information by people without visual impairments.

## 2.4. Haptic devices

Haptic devices are used to provide tactile feedback during user interaction with virtual 3D models by generating opposing forces with the help of servomotors or ultrasound [68] by changing the air pressure around the user with the help of specially designed ultrasound speakers. As a new direction for future research, also a plasma-generated haptic sensation has been reported in [59] when the user touches the plasma voxels generated by their femtosecond laser system.

Due to the importance of having a real-time tactile response, haptic devices are used by numerous research teams around the world and are representing a valuable tool during the training of medical students. Surgeons are able to plan and exercise complex operations on virtual models of their patients like in the case of the Selman Surgery Rehearsal Platform [69] developed by neurosurgery professor Warren Selman from the University Hospital of Cleveland, Ohio, together with Moty Avisar and Alon Geri two former Israel Air Force officers [70]. This neurosurgery simulator uses 3D images generated from previous CT and MRI scans of the patients (as shown in Figure 2.29) and enables observation of the interaction between the

virtual tissues and a set of surgical instruments, resulting an experience that is really close to a real-life surgical intervention.



Figure 2.29. Selmán Surgery Rehearsal Platform

(source: <http://www.geomagic.com/files/7613/4817/7685/Surg-Theater-SRP-with-UH-Selman-04-02-2012.png>)

The Selmán Surgery Rehearsal Platform uses two connected Geomagic Touch haptic devices [71] that applies force feedback on the user's hands and provides a tactile sensation similar to the interaction with a living tissue during virtual operations, assuring an unprecedented experience for future surgeons.

MedaPhor [72] developed a virtual ultrasound training simulator named ScanTrainer that is using real ultrasounds of patients that enables the trainees to study more advanced medical cases in a non-clinical environment as shown in Figure 2.30.



Figure 2.30. ScanTrainer

(source: <http://www.geomagic.com/files/2913/4817/8003/curriculum-based.png>)

The interaction with the displayed 3D models is achieved through a Geomagic Touch haptic device and the application also provides an evaluation module for the trainees.

## 2.5. Conclusions

The aim of this chapter is to provide an overview of the latest technological innovations that any researcher needs to consider when faced with building a state-of-the-art user interface. Besides the gesture-based interfaces that assure a natural user interaction with the latest 3D display technologies, we need to consider the possibilities offered by BCIs in cases that the targeted users are limited in achieving their goals by various medical conditions.

The LEAP Motion controller is very accurate in detecting individual fingers at a very high detection rate, which is surpassed only by the Google's Soli project that offers an even smaller device suitable for wearable interfaces. As an alternative gesture tracking technology for mobile devices the SideSwipe and SoundWave technologies can provide a more convenient solution.

While the MYO armband's development purpose is to detect various gestures by interpreting electrical signals at the muscular level, it is also suitable to control a prosthetic limb (as previously described by the author of the thesis) thus becoming also a BCI. The main advantage of MYO over the OHMG is that it doesn't require an invasive surgery, making it less prone to clinical injuries during operation. However the OHMG is recommended for achieving a better signal.

The EPOC headset offers a non-invasive alternative to the BrainGate technology where the microelectrode array needs to be surgically implanted to the patients in order to receive the electrical signals directly from the brain. While these two technologies are reading signals, the FES is used to stimulate the muscles in order to avoid muscular atrophy and help the patients to have a more active lifestyle.

Stereoscopic displays provide the stereopsis (the 3D depth perception) with the help of various 3D glasses. While this technology provides a single perspective, the autostereoscopic displays (as previously shown) are capable to provide several different perspectives to multiple users without the use of any kind of special additional equipment. Some of the enhanced versions of autostereoscopic displays are providing also 3D volumetric data that takes the form of individual voxels or hogels depending on the image element's positioning. In some special cases the surrounding air (or thin water vapors) can be also used as projection surface (e.g. Heliodisplay).

With the purpose of providing a tactile feedback for the blind persons several Braille terminals were developed that are using a pattern of rounded pins to simulate visual feedback. This technology has been further developed into haptic devices (e.g. Geomagic Touch) to assure tactile feedback that is not available otherwise during the visualization of 3D content.

Based on the researchers specific needs these technologies can be used separately or combined in order to provide a more elaborate view of the studied 3D models and to configure specific user interaction scenarios.

The Table 2.1 presents a comparison of the hardware technologies presented in the current chapter organized by devices, category, characteristics and recommended use.

Table 2.1 Hardware technologies comparison

<b>Devices</b>	<b>Category</b>	<b>Characteristics</b>	<b>Recommended use</b>
LEAP Motion	gesture recognition	<ul style="list-style-type: none"> <li>- uses a pair of infrared cameras to extract 3D depth information</li> <li>- can detect the motion of individual arms, hands and fingers with sub-millimeter accuracy</li> <li>- detection rate of 200 frames per second</li> <li>- doesn't require any prior camera calibration</li> </ul>	In cases where the tracking of individual fingers is essential and multiple users are using the same device.
MYO Armband	gesture recognition	<ul style="list-style-type: none"> <li>- equipped with medical grade stainless steel EMG sensors</li> <li>- nine-axis IMU containing three-axis gyroscope, three-axis accelerometer, three-axis magnetometer</li> <li>- it requires prior calibration</li> </ul>	In cases where the researchers need to detect and interpret the electrical activity in the user's muscles (e.g. to control a prosthetic limb).
SideSwipe	gesture recognition	<ul style="list-style-type: none"> <li>- uses the actual GSM signal to detect in-air hand gestures both above and around a mobile device</li> <li>- capable to distinguish between three categories of</li> </ul>	For gesture-based interaction with the available mobile devices based on the GSM signal.

		gesture sets such as tap, hover and slide	
SoundWave	gesture recognition	- uses the speaker and microphone of a computer system to detect in-air gestures	For gesture-based interaction with the available mobile devices based on the Doppler effect.
Soli	gesture recognition	- low-power radar sensor - operates in the 60 GHz ISM (industrial, scientific and medical) radio band - very high frame rate (from 100 to 10000 frames per second)	Due to the small size of the chip it represents a suitable technology for inclusion into future wearable devices.
EPOC+ headset	BCI	- is a non-invasive neural interface - 14 EEG sensors and 2 references for reading the electrical signals produced by the user's brain - nine-axis IMU - Bluetooth SMART 4.0 LE connection	In cases where invasive surgery needs to be avoided.

FES	BCI	<ul style="list-style-type: none"> <li>- is a technique of applying controlled electrical impulses to the muscles of individuals who suffered a SCI</li> <li>- the electrodes can be applied on the skin, above the paralyzed muscles, or surgically inserted directly on the nerves that communicate with the muscles</li> </ul>	For patients who suffered a SCI (Spinal Cord Injury).
OHMG	BCI	<ul style="list-style-type: none"> <li>- allows bidirectional communication between the attached prosthetic limb and the surgically implanted electrodes in the muscles, the patient being able to have a tactile perception</li> </ul>	In cases where there is necessary to transmit the signals between the peripheral nervous system and the prosthetic limb.
BrainGate	BCI	<ul style="list-style-type: none"> <li>- 4x4 mm, 96-channel microelectrode array implanted in the motor cortex</li> </ul>	Recommended for patients with severe paralysis.
Stereoscopic displays	display technology	<ul style="list-style-type: none"> <li>- requires special glasses in order to provide the 3D depth perception</li> </ul>	Used in movie theaters where a single view is enough to show the 3D content.
Autostereoscopic (multiview) displays	display technology	<ul style="list-style-type: none"> <li>- do not require the use of any kind of special</li> </ul>	In cases where a single display needs to display two types of content at the

48 2. Hardware Technologies Overview

		glasses - provides a finite number of different perspectives of the same 3D object to a finite number of viewers	same time.
Volumetric 3D displays	display technology	- are using individual voxels - a 360 degree view of the rendered 3D model	In cases where multiple users need to see the same 3D content from different angles.
Pseudo-holographic displays	display technology	- are displaying individual hogels - the 3D image seems to float in midair that resembles to an actual hologram	Initially used for entertainment and comercial purposes. It has a great potential for scientific applications.
Holographic film projections	display technology	- uses a special holographic film that contains a large number of lenticular lenses	It is very suitable for telepresence and marketing applications.
Heliodisplay	display technology	- is an air-based interactive display	The display uses dry air and helps avoiding the damage of other electronic components.
Braille display	display technology	- the characters are formed by a series of rounded pins that are grouped and raised in patterns that are detected by the users with their finger tips	To display Braille characters for blind users who cannot use a regular display.



Geomagic Touch	haptic devices	- the device is generating opposing forces with the help of servomotors.	Are used to provide tactile feedback during user interaction with a virtual 3D model.
----------------	----------------	--	---

As it will be presented further in the thesis, mastering the advancements in technology paves the way towards a better visualization and manipulation of computer generated pseudo-holographic 3D content.

### 3. SOFTWARE TECHNOLOGIES. OVERVIEW AND CONTRIBUTIONS

The current chapter describes a set of open source software technologies used during various phases of the research. The aim and the major contribution brought by the applications developed based on these libraries is to demonstrate that researchers can achieve viable software alternatives that do not depend on specific pre-installed operating systems or proprietary software components. This is considered to be an important aspect that can affect the budget and the visibility of many research projects as discussed in [7] and [8].

Throughout the course of the research the author of the thesis has proposed several software architectures that are meant to support the daily work of medical professionals, facilitate the learning process for medical students and that also can be adapted to the fields of bioinformatics or chemistry as it will be shown in a later chapter.

The first theoretical proposal [73] was a medical support system for diagnosing patients during trauma recovery, which can be developed using standards like WebRTC [11] for real-time communication between physicians and several open source software libraries such as JSARToolKit [12], ThreeJS [13] and NanoDICOM [14]. This system could use the patients DICOM files to create a 3D reconstruction of the studied anatomical area (using NanoDICOM and ThreeJS) that would be superimposed on the image of the real person with the help of JSARToolKit creating an augmented layer that could be compared against the real evolution of the studied affection that the patient has. This virtual assembly could be shared with remote colleagues via WebRTC for an increased accuracy during the examination of the symptoms that might require a second opinion of a specialist. This augmented reality based approach would allow the rotation of the virtually attached 3D model in the same time as the subject rotates (e.g. his arm) providing a more accurate 3D view of the studied area to the medical staff during investigation as shown in Figure 3.1.

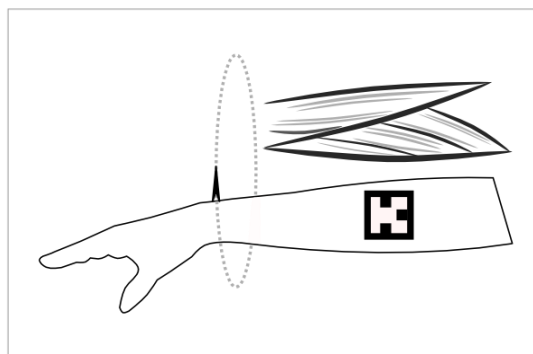


Figure 3.1. The virtually attached 3D model rotates in the same time as the real patient's arm presented by the author in [73]

Similar software architectures have been proposed by other research teams in [3], [4] and [8] however these are not completely web-oriented, imposing various kinds of limitations during installation. In order to overcome these limitations several other solutions have been developed by various teams as identified by the author of this thesis in [74]. The significance of investigating this new type of open source imaging applications has been underlined by other research teams as well in [75] and [76].

The second approach was to implement the results obtained during the study of the first theoretical module. This allowed the adaptation of the proposed system to gesture tracking devices (e.g. a LEAP Motion Controller) for gesture-based control of the rendered 3D model that is displayed both on a regular 2D monitor and a custom-built 3D pseudo-holographic display for an enhanced visual experience during interaction.

As a third option, to demonstrate the versatility of such an open-source based system, the application was adapted to GLmol [17], a browser-based 3D molecular viewer developed by Takanori Nakane [77]. The GLmol application was extended by the author of the thesis by implementing gesture-based interaction with the rendered molecules with the help of a LEAP Motion Controller. Since GLmol's main code is written in JavaScript it was a suitable option to add the gesture-tracking functionalities provided by LeapJS [18] that could be interfaced with the application using ThreeLeapControls [19].

### 3.1. NanoDICOM

NanoDICOM [14] is a PHP DICOM parser that is released under the MIT License. The latest stable version 1.3 is called Ancient Cajamarca and has been released in 2011. The toolkit is available for download from GitHub at the following address <https://github.com/nanodocumet/Nanodicom/tree/v1.3>. Since this is a server-side parser it requires at least PHP version 5.2.6 that can be installed on a server under a Windows7, Linux (tested by the development team on Debian Lenny and Ubuntu 10.04), Solaris and Mac OS X operating systems.

According to the project's page mentioned above, NanoDICOM supports a large number of Transfer Syntaxes that are enumerated in Table 3.1. This Transfer Syntaxes are a set of individual encoding rules that are used to describe Abstract Syntaxes, and each have a corresponding Unique Identifier (UID) well defined in the DICOM standard [78].

Table 3.1 Supported Transfer Syntaxes by NanoDICOM

<b>Transfer Syntax</b>	<b>Unique Identifier (UID)</b>
Implicit VR Little Endian (The Default Transfer Syntax)	1.2.840.10008.1.2
Explicit VR Little Endian	1.2.840.10008.1.2.1
Deflated Explicit VR Little Endian	1.2.840.10008.1.2.1.99
Explicit VR Big Endian	1.2.840.10008.1.2.2
JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 bit Image Compression	1.2.840.10008.1.2.4.50

JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 bit Image Compression (Process 4 only)	1.2.840.10008.1.2.4.51
JPEG Lossless, Non-Hierarchical (Process 14)	1.2.840.10008.1.2.4.57
JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression	1.2.840.10008.1.2.4.70
JPEG 2000 Image Compression (Lossless Only)	1.2.840.10008.1.2.4.90
JPEG 2000 Image Compression	1.2.840.10008.1.2.4.91
RLE Lossless	1.2.840.10008.1.2.5

NanoDICOM doesn't provide a separate online viewer and in the first theoretical approach it was used by the author only to provide a lightweight, open source alternative to extract the imagistic data from the patients' DICOM files and save them on the server as individual PNG slices. These PNG files afterwards would be used as textures for the multiplanar 3D reconstruction on the client-side with the help of the ThreeJS component. The steps required to obtain the final result are summarized in Table 3.2.

Table 3.2 The steps required to obtain a 3D image

Steps	Description
1	The user uploads a DICOM file to the server
2	The NanoDICOM component extracts the imagistic data and saves it as a PNG file on the server
3	Each PNG file is sent back to the user's browser
4	The PNG files are used to reconstruct the 3D model of the anatomical area with the help of the ThreeJS component

Due to its robustness and speed, NanoDICOM represents a suitable alternative for developing an online DICOM viewer, that can be used by physicians to collaborate with other specialists when they need a second opinion.

### 3.2. JSARToolKit

JSARToolKit [12] is a JavaScript software library for building Augmented Reality applications written by Ilmari Heikkinen [79] and released under the GPL 3 license in 2011. The library represents an indirect adaptation of the ARToolKit [80] library that was initially developed in the C programming language by Hirokazu Kato. In order to offer a greater flexibility to developers ARToolKit has also been ported to a Java version, resulting in NyARToolKit [81] and to an ActionScript 3 version named FLARToolKit [82]

All versions of ARToolKit are used to calculate the distance from the camera to a physical marker attached to a real object in real time by analyzing each individual frame.

The type of the marker used by JSARToolKit is also often referred as an ID marker due to the fact that it allows the association of a specific ID number with it at the software level. This pairing allows the developers to track multiple markers at the same time inside the video feed and to virtually attach various 3D objects to each particular marker that is detected. The physical ID markers need to have a specific pattern in order to be detected by the software and they must have a bold black contour (as we can see in Figure 3.2) that helps in delimiting them from the rest of the scene.

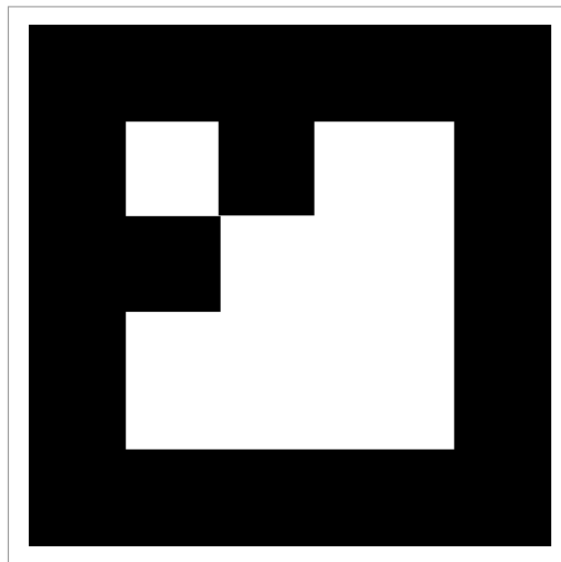


Figure 3.2. Example of an ID marker

The pattern inside the bold contour needs to be simple since this makes it easier to track the movement of each marker across each frame provided by the camera, unlike the QR codes where the displayed pattern is interpreted only once for retrieving the associated data (e.g. the URL of a website).

The markers also have the role of a reference point across frames and are helping to determine the orientation and the position of the virtually attached 3D model as we can see in Figure 3.1. This functionality permits the scaling and the rotation of the rendered 3D objects, assuring that they will always be at their relative position towards the associated real object.

In the proposed solution a real marker would have been attached with a self-adhesive band to the patient and afterwards the generated 3D reconstruction of the studied area would have been superimposed inside the video feed resulting in a augmented view of the reality.

JSARToolKit uses a HTML5 *canvas* element for detecting the markers and draws each video frame on it in the order that they are retrieved from the video stream of the camera. To achieve this functionality the application uses the

*drawImage()* method available in any modern browser through its API (if implemented) and loops through the video frames with the help of the *requestAnimationFrame()* method.

The JSARToolKit represents an important step towards achieving complete independence towards specific operating systems and allows a greater portability of the future web applications, eliminating the need for any third-party browser plug-ins.

### 3.3. CornerstoneJS

CornerstoneJS [83] is a set of JavaScript libraries used for visualizing medical images (that are retrieved after parsing the patients' DICOM files) with the help of the HTML5 *canvas* element. The libraries are written by Chris Hafey and are released under the open source MIT license starting from 2014.

Unlike NanoDICOM, the Cornerstone libraries allow a pure client-side implementation of a more complex medical application, since there is no need to send each DICOM file to a remote server and retrieve the results after parsing.

The core module of Cornerstone needs an *ImageLoader* plug-in to load the images and this is provided as a separate component called *cornerstonWADOImageLoader* [84]. This library implements the DICOM Web Access to DICOM Objects (WADO) standard [85] that allows the retrieval of DICOM objects from a remote DICOMWeb server. The various Transfer Syntaxes supported by the *cornerstonWADOImageLoader* are listed in Table 3.3.

Table 3.3 Supported Transfer Syntaxes by *cornerstonWADOImageLoader*

Transfer Syntax	Unique Identifier (UID)
Implicit VR Little Endian (The Default Transfer Syntax)	1.2.840.10008.1.2
Explicit VR Little Endian	1.2.840.10008.1.2.1
Deflated Explicit VR Little Endian	1.2.840.10008.1.2.1.99
Explicit VR Big Endian	1.2.840.10008.1.2.2
JPEG Baseline (Process 1-8 bit): Default Transfer Syntax for Lossy JPEG 8 bit Image Compression	1.2.840.10008.1.2.4.50
JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 bit Image Compression	1.2.840.10008.1.2.4.51
JPEG Lossless, Nonhierarchical (Process 14)	1.2.840.10008.1.2.4.57
JPEG Lossless, Nonhierarchical, First- Order Prediction (Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression	1.2.840.10008.1.2.4.70
JPEG-LS Lossless Image Compression	1.2.840.10008.1.2.4.80
JPEG-LS Lossy (Near-Lossless) Image Compression	1.2.840.10008.1.2.4.81

JPEG 2000 Image Compression (Lossless Only)	1.2.840.10008.1.2.4.90
JPEG 2000 Image Compression	1.2.840.10008.1.2.4.91
RLE Lossless	1.2.840.10008.1.2.5

This separation of the *ImageLoader* module allows the developers to write their own libraries if they want to extend the functionalities of their application by loading even proprietary medical image formats in case they need to do so. The *cornerstoneWADOImageLoader* even if it can retrieve pixel data from a remote server, it also makes it possible to load raster images from local files as we will see later in the thesis.

The parsing of the DICOM files is done by a separate JavaScript library named *dicomParser* [86] that is used only to retrieve the information stored in the files. This module can extract all the related data that is stored in the files during a CT or MRI scan, such as: patient's age, name, gender, weight, acquisition time, slice thickness, magnetic field strength, device manufacturer, software version, last calibration date, pixel spacing, window width, window center, etc.

Cornerstone also provides a set of tools that can be used to interact with the displayed images, all contained in the *CornerstoneTools* [87] library. The basic functionalities provided by the library are:

- adjustment of window width and window center;
- image scaling;
- image panning;
- selection of a rectangular region of interest;
- selection of an elliptical region of interest;
- taking a pixel probe;
- angle measurements;

The author of the thesis used Cornerstone as part of a more comprehensive example of a medical visualization system that will be later detailed in the thesis.

### 3.4. HTML5

The current major revision of the HyperText Markup Language, HTML5, was released as a stable *Recommendation* on the 28th of October 2014 [88], followed by a minor revision 5.1 published on the 1st of November 2016 by the W3C (World Wide Web Consortium) [89]. The standard has been developed jointly by the W3C and the WHATWG (Web Hypertext Application Technology Working Group), a working group founded by employees of major browser vendors such as Apple, Mozilla and Opera.

HTML5 brings some major changes described in [90], a draft document published as a *Working Group Note* by the W3C. However the new HTML standard has been defined in such a way that it is backward compatible, so the user agents (e.g. web browsers) will support the older elements and attributes that are no more part of the current specification. This is a key aspect that had to be considered because web browsers will need to be able to display web pages in the future that have been written using prior versions of the HTML5 standard. Even if the browser vendors need to provide support for these obsolete elements, web developers are no more allowed to write their future web applications using deprecated elements. Moreover in order to keep the markup language simple for developers, various older

elements and attributes that were related to the presentational aspect, in the new version are taken care of by a separate CSS (Cascading Style Sheets) style sheet. This approach facilitates the separation of the document content from the presentation (e.g. font's colors, text sizes, various elements' padding, etc.).

Numerous new elements have been introduced in HTML5 for a better document structure and making clear their semantic role for web developers. The Table 3.4 summarizes the most important new elements introduced by the specification.

Table 3.4 Important new elements introduced in HTML5

<b>Element's name</b>	<b>Description</b>
<i>article</i>	Depicts a reusable piece of content inside a web page such as an online newspaper article, forum post, users' comment or blog entry.
<i>section</i>	Used to represent a thematically identifiable section of a web page that can be specified by including a heading element ( <i>h1</i> , <i>h2</i> , <i>h3</i> , <i>h4</i> , <i>h5</i> or <i>h6</i> ) that clearly indicates the structure of the page.
<i>main</i>	Constitutes the main content of the <i>body</i> element of a web page or application and should be related to the main topic of the page or the main functionality offered by the web application.
<i>aside</i>	Unlike the <i>main</i> element, <i>aside</i> represents a part of the web page that is only partially related to the main topic (e.g. advertising or quotes inside a blog).
<i>header</i>	Mostly used to delimitate introductory content (e.g. a section's heading element).
<i>footer</i>	Used as a footer for a <i>section</i> or the <i>body</i> of the web page, usually stores information about the author or related documents. There might be more than one <i>footer</i> elements in a document, each associated with its own ancestor.
<i>nav</i>	Defines a section that contains navigation links to other pages or other sections in the same document.
<i>figure</i>	Used to annotate images, diagrams or sketches inside a web page.
<i>template</i>	Used to declare clonable HTML fragments that can be manipulated at runtime by scripts.
<i>bdi</i>	Allows the bidirectional formatting of text inside a document.
<i>video</i>	Used to display video or movie content inside a web page.
<i>track</i>	This element provides a text track for a <i>video</i> element, when subtitles, captions, descriptions or other metadata is available.
<i>audio</i>	Used to play audio content inside a document.
<i>source</i>	This element is used to define multiple sources for a <i>video</i> or <i>audio</i> element.
<i>canvas</i>	Used to render graphics on the fly with the help of JavaScript (or other scripts).



<i>embed</i>	Acts as a container for external plugins.
<i>progress</i>	This element can be used to indicate the progress of a specific time consuming operation (e.g. downloading a file).

Other elements have become obsolete in HTML5 for various reasons, they either are impairing usability or they had a presentational scope (now handled by CSS). The Table 3.5 reflects the obsolete elements in HTML5.

Table 3.5 Obsolete elements in HTML5

<b>Obsolete element</b>	<b>Recommended to use instead</b>
<i>acronym</i>	<i>abbr</i>
<i>applet</i>	<i>embed</i> or <i>object</i>
<i>dir</i>	<i>ul</i>
<i>isindex</i>	<i>form</i> and text field combination
<i>bgsound</i>	<i>audio</i>
<i>frame</i>	<i>iframe</i> or CSS
<i>frameset</i>	<i>iframe</i> or CSS
<i>noframes</i>	<i>iframe</i> or CSS
<i>listing</i>	<i>pre</i> and <i>code</i>
<i>noembed</i>	<i>object</i>
<i>strike</i>	<i>del</i>
<i>basefont</i>	CSS
<i>big</i>	CSS
<i>blink</i>	CSS
<i>center</i>	CSS
<i>font</i>	CSS
<i>marquee</i>	CSS
<i>multicol</i>	CSS
<i>nobr</i>	CSS
<i>spacer</i>	CSS
<i>tt</i>	CSS

With the purpose of a greater modularity some of the newly developed APIs included initially in HTML5 became part of separate specifications published by the W3C and their development has been deferred to various working groups. These new technologies provide a great resource for web developers allowing them to build more complex web applications and cross-platform mobile applications. Some of the most important newly provided APIs are:

- *File API* – allows the access of web applications to local files;
- *Drag and drop API* – allows the dragging and dropping of items on web pages (e.g. when opening a file from the local filesystem);
- *Web Socket API* – allows full-duplex communication between a web page and the server;
- *Server-sent events* – allows a server to send data to the client (e.g. an automatic update) after an initial connection has been established without a specific request from the client;
- *Geolocation API* – provides information regarding the geographical location of a clients' device (e.g. by using Location Information Servers);

- *WebRTC* – allows direct real-time communication between two browsers without the installation of additional plugins;
- *XMLHttpRequest API* – allows asynchronous update of only a section of a document without a full page refresh, providing a more dynamic experience. This standard is often used in AJAX (Asynchronous JavaScript and XML) for data transfer between the client application and a server;
- *Web storage* – allows the storage of data by browsers;

Due to the wide range of technologies offered by HTML5, the author of the thesis used it during the development of all the applications during the research as it will be detailed later in the next chapters.

### 3.5. WebRTC

WebRTC is a set of open source APIs and protocols that allows Real-Time Communication between browsers (often referenced as peers) without using any third-party software components or any additional plug-ins. The framework is being actively developed by the W3C on the API level [91] together with the IETF (Internet Engineering Task Force) on the protocol level [92]. There is also an alternative reference implementation called OpenWebRTC [93] released under the BSD-2 license, which provides a flexible, mobile-first WebRTC client framework for building native mobile apps.

At this moment WebRTC is supported by the major browser vendors, like Google, Mozilla, Microsoft (only Edge) and Opera and it mainly comprises the following JavaScript APIs:

- *MediaStream* - that allows the browser to access the camera or microphone for capturing media streams with *getUserMedia()*;
- *MediaRecorder* – used for recording audio or video streams with the help of *MediaStreams*;
- *RTCPeerConnection* – allows the streaming of audio or video between two peers of a connection (e.g. one local and one remote browser) without an intermediary server;
- *RTCDataChannel* - facilitates the direct streaming of data between two peers;

WebRTC coordinates the communication and sends control messages between peers by using a control mechanism known as signaling that is not part of the specification. This allows the developers to have a greater flexibility when they choose the messaging protocol they want to use. As specified in [11] signaling has the purpose to communicate the following information between peers:

- *Session control messages* - that have the role to start and close the communication between the peers, eventually reporting the errors that might have occurred;
- *Network configuration* – used to communicate to the other peer the computer's IP addresses and ports available for communication;
- *Media capabilities* – exchanges information regarding the codecs and resolutions available by each peer;

Due to the possibilities offered by WebRTC, the author of the thesis proposed a medical support system [73] that would facilitate the communication between physicians during the investigation of complex medical cases.

Some concrete implementations of WebRTC are the browser-based video chat web application provided by vLine [94] and Sharefest [95], a web application that supports the sharing of files from the local computer to a remote peer without additional plugins.

### 3.6. WebGL

WebGL (Web Graphics Library) [96] is a cross-platform, low-level JavaScript 3D graphics API, derived from OpenGL ES 2.0 (OpenGL for Embedded Systems), and used to render 2D and 3D graphics inside a web page eliminating the need for third-party plug-ins to accomplish this task. The library is actively developed and maintained by the Khronos WebGL Working Group that involves contributions from all the major browser vendors such as Google, Apple, Mozilla and Microsoft.

WebGL relies on the HTML5 *canvas* element that is used to draw interactive 3D graphics within a web page. The canvas is accessed by JavaScript that handles the application logic, while WebGL is used to rasterize primitives such as points, lines or triangles directly accessing the GPU (Graphics Processing Unit). The code that is run on the GPU comes in the form of a pair of functions called *shaders* that are written in GLSL (OpenGL Shading Language). There are two types of shaders: one is the vertex shader, that computes vertex positions, and the other is the fragment shader, that computes the color of each pixel used to draw a primitive (e.g. a point, a line or a triangle).

Due to the complexity of the code generated by WebGL some other libraries have been developed that already have components for rendering a scene, a camera, a light source, various shapes, materials or textures (e.g. ThreeJS, SceneJS, GLGE, PhiloGL).

### 3.7. ThreeJS

ThreeJS [97] is a JavaScript API used for rendering interactive 3D models within a web page without using third-party plug-ins. The ThreeJS library was initially released on GitHub in 2010 by Ricardo Cabello under the MIT license and has almost 800 contributors to the source code since then.

The library significantly reduces the size of the code that needs to be written usually using WebGL. Besides the WebGLRenderer object, the ThreeJS API has optional renderers such as CanvasRenderer and SVGRenderer in case that WebGL is not supported by the browser, allowing a greater flexibility to application developers.

ThreeJS simplifies the generation of complex 3D scenes inside a browser-based application since it provides a large number of readily-available components through its API:

- *Scene* – one of the key components used to display a 3D model inside a web page;
- *Cameras* - used to view the 3D models rendered inside the scene;

- *Renderers* - used to draw the 3D models inside the scene;
- *Lights* - used to lightup the scene and the objects inside of it;
- *Materials* - used to change the properties of the rendered 3D objects;
- *Textures* - defines the surface appearance of the displayed objects;
- *Loaders* - to load various file types obtained from other 3D computer graphics software (e.g. Blender, Maya);

The author of the thesis used the ThreeJS library for the multiplanar 3D reconstruction of the studied anatomical area inside the application. For the 3D reconstruction the application uses the 2D PNG files obtained from the patient's DICOM slices applied as textures to a set of *PlaneGeometry* objects returned by the *THREE.PlaneGeometry()* method. The process will be further detailed in the next chapter.

### 3.8. LeapJS

LeapJS [18] is a JavaScript library used for tracking gestures inside a web application with the help of a Leap Motion Controller. The tracking is done by two inbuilt cameras that are detecting the infrared light with a wavelength of 850 nm provided by three infrared LEDs for a better coverage. Even if the light emitted by the LEDs is outside the visible spectrum it can be seen with the help of a camera as shown in Figure 3.3. This demonstrates that the LEAP Motion doesn't project any infrared dot pattern as Kinect does by using structured light to infer the shape of an object.



Figure 3.3. The IR lights of the LEAP Motion Controller

The field of view of the stereo camera pair has an approximate shape of an inverted pyramid that spans 60 cm above the device. The accuracy and the range of detection is influenced by the light intensity emitted by the LEDs since there is a limited maximum current that the LEAP Motion Controller can draw through the USB port. The detection is also influenced by external light sources that might emit infrared light such as light bulbs or the Sun, and it's important to avoid the direct exposure of the controller to these.

On the software side, the device provides raw sensory data regarding the interaction space through the Leap Service that is basically a locally installed WebSocket server. This service sends a *Frame* object every 10 ms over the WebSocket to the client web application that uses LeapJS to extract information from it. The *Frame* object is defined in the API and represents a blob of JSON data containing the raw information regarding the position of each individual hands, fingers and finger-like tools detected in a single frame. The data streamed by the cameras is matched against an internal model of the human hand or tools thus providing a very precise tracking result. The predictive tracking algorithm of the device also applies some filtering techniques to infer the position of the occluded fingers to estimate their accurate positions.

Due to its robust detection capabilities and the openly available LeapJS API the author of the thesis used the LEAP Motion device in both applications that will be detailed further in the next chapters.

### 3.9. ThreeLeapControls

The ThreeLeapControls [19] JavaScript library was written by Torsten Sprenger in 2013 and represents an interface between LeapJS and ThreeJS used for mapping the individual positions in space of the detected hands and fingers to their corresponding 3D positions inside the rendered 3D scene.

The library provides two separate options for interacting with a 3D virtual object (using LeapObjectControls) or to interact with the virtual camera used to visualize the 3D objects inside the scene (with the help of the LeapCameraControls).

With the help of ThreeLeapControls, application developers can set the number of fingers and hands that need to be recognized by the LEAP Motion Controller in order to trigger specific actions such as scaling, rotating or panning of the rendered 3D objects inside the scene and rotating, zooming or panning the camera object for a more complex interaction with the scene. All the controls provided by ThreeLeapControls can be individually enabled or disabled by simply setting the associated parameter to true or false inside the source code. The other options provided by the library include:

- adjusting the speed of each specific control;
- setting the number of hands that need to be detected in order to trigger a specific action;
- possibility to specify which individual fingers need to be recognized to trigger the associated actions;
- setting the minimum and maximum distances allowed for each action;
- distinguish between left or right hand used to complete a specific action;

Due to the fact that ThreeLeapControls uses LeapJS v 0.2.0 and ThreeJS rev. 59, the author of the thesis had to update it to the latest functionalities

provided by the current LeapJS v 0.6.4 and ThreeJS rev. 79. The newly contributed code detects the number of fingers that are extended (and which fingers are extended) in order to provide functionalities such as rotate, scale and pan for each tracked object and rotate, zoom and pan for each camera controlled.

### 3.10. Conclusions

The presented open source libraries and APIs are providing a valuable resource for every development team faced with the challenge to provide a browser-based medical imaging application. These software solutions have a large applicability in other areas where gesture-based interaction facilitates the interaction with complex 3D models as further demonstrated in the thesis.

The NanoDICOM toolkit allows the extraction of relevant imagistic data from the DICOM files but it needs to be installed on the server side increasing the execution times of the application since the extracted data needs to be sent back to the browser for visualization. A better alternative is provided by CornerstoneJS (written in JavaScript) that runs inside a web browser and has also an image loader component that allows the manipulation of the extracted images. Both solutions can harvest the features of WebRTC that allows direct communication between two peers (browsers) without installing additional plug-ins or third-party software components, thus decreasing the application costs and simplifying its architecture.

While JSARToolKit needs a physical marker to be attached to a real world object that needs to be tracked across the video frames, ThreeLeapControls can provide a similar functionality without a marker by linking the hands position extracted with LeapJS to the rendered 3D virtual object (using ThreeJS) inside a 3D scene provided by WebGL on a HTML5 *canvas* element.

The use of the LeapJS library totally eliminates any extra calibration steps that are usually needed in other environments where multiple cameras are used to extract 3D depth related information coming from multiple users of the application. This is a major improvement because multiple users can interact with the application at the same time without having to undergo a separate training phase that is needed in older applications to differentiate between the shapes of the hands of each particular individual.

The Table 3.6 presents a comparison of the software technologies described in the current chapter highlighting their characteristics and recommended use for various research projects.

Table 3.6 Software technologies comparison

Technology	Characteristics	Recommended use
NanoDICOM	<ul style="list-style-type: none"> <li>- PHP DICOM parser</li> <li>- latest stable version 1.3 is called Ancient Cajamarca</li> <li>- requires at least PHP version 5.2.6</li> <li>- supports a large number of Transfer Syntaxes</li> </ul>	Parsing DICOM files on the server-side.

JSARToolKit	<ul style="list-style-type: none"> <li>- JavaScript software library for building Augmented Reality applications</li> <li>- allows the association of a specific ID number with a marker at the software level</li> </ul>	Used to calculate the distance from the camera to a physical marker attached to a real object in real time by analyzing each individual frame.
CornerstoneJS	<ul style="list-style-type: none"> <li>- JavaScript DICOM parser</li> <li>- supports a large number of Transfer Syntaxes</li> <li>- allows the retrieval of remote DICOM objects</li> <li>- has several modules used for image manipulation</li> </ul>	Parsing DICOM files on the client-side (with the dicomParser component).
HTML5	<ul style="list-style-type: none"> <li>- current major revision of the HyperText Markup Language</li> <li>- has several new elements that assure a better document structure</li> <li>- provides API's for accessing local files, drag and drop functionality, full-duplex communication between a web page and the server, geolocation, asynchronous update of documents and real-time communication between two browsers</li> </ul>	For building complex web and cross-platform applications.
WebRTC	<ul style="list-style-type: none"> <li>- a set of APIs and protocols that allows Real-Time Communication between peers</li> <li>- supported by the major browser vendors</li> <li>- coordinates the communication between browsers by using a control mechanism known a signaling</li> </ul>	For browser-based video chat or file sharing web applications.
WebGL	<ul style="list-style-type: none"> <li>- a JavaScript 3D graphics API, derived from OpenGL ES 2.0</li> </ul>	For rendering 2D and 3D graphics inside a web page (eliminates the need for third-party plug-ins).

	<ul style="list-style-type: none"> <li>- relies on the HTML5 <i>canvas</i> element that is used to draw interactive 3D graphics within a web page</li> </ul>	
ThreeJS	<ul style="list-style-type: none"> <li>- significantly reduces the size of the code that needs to be written (usually using WebGL)</li> <li>- WebGLRenderer object</li> <li>- CanvasRenderer object</li> <li>- SVGRenderer object</li> </ul>	For rendering interactive 3D models (objects) within a web page (without using third-party plug-ins).
LeapJS	<ul style="list-style-type: none"> <li>- a JavaScript library</li> <li>- extracts the raw sensory data provided by the LEAP Motion regarding the interaction space</li> </ul>	For tracking gestures inside a web application with the LEAP Motion Controller.
ThreeLeapControls	<ul style="list-style-type: none"> <li>- a JavaScript library</li> <li>- provides two separate options for interacting with a 3D virtual object (using LeapObjectControls) or to interact with the virtual camera used to visualize the 3D objects inside the scene (with the help of the LeapCameraControls)</li> <li>- adjusting the speed of each specific control</li> <li>- setting the number of hands that need to be detected</li> <li>- distinguish between left or right hand</li> </ul>	For mapping the individual positions in space of the detected hands and fingers to their corresponding 3D positions inside the rendered 3D scene.

The open source cross-platform solutions outlined in this chapter are facilitating the addition of other output devices, such as optical head-mounted or pseudo-holographic displays when extra functionalities are needed.

In the next chapter the author of the thesis will demonstrate how these software technologies can work together with the result of a more complex medical imaging application that implements both a classical 2D user interface and a modern 3D interface where the user can interact with the displayed 3D model using various hand gestures.



## 4. DEVELOPMENT OF A BROWSER-BASED DICOM VIEWER

The current chapter describes the development of a browser-based DICOM viewer that provides gesture-based interaction with the rendered 3D model of the studied anatomical area without using any additional browser extensions or third-party plug-ins. Even if there are plenty of open source libraries available this is still not a trivial task because of the limitations regarding the execution time of the scripts on the client-side of a web-oriented software application. The author of the thesis had to provide a simplified segmentation algorithm [98] that is capable to extract the pixel data from the DICOM files in a timely manner, before the browser stops the script execution. This is currently limited in Mozilla Firefox to 10 seconds, in Safari to 5 seconds, while in Internet Explorer to five million statements executed.

The application named simply JSDV (JavaScript DICOM Viewer) is based on the open source libraries described in chapter 3. In order to extract the imaging information from the DICOM files the author used the CornerstoneJS library.

Starting from the developer's application example found at <https://rawgit.com/chafey/cornerstoneWADOImageLoader/master/examples/dicomfile/index.html> as we can see in Figure 4.1,

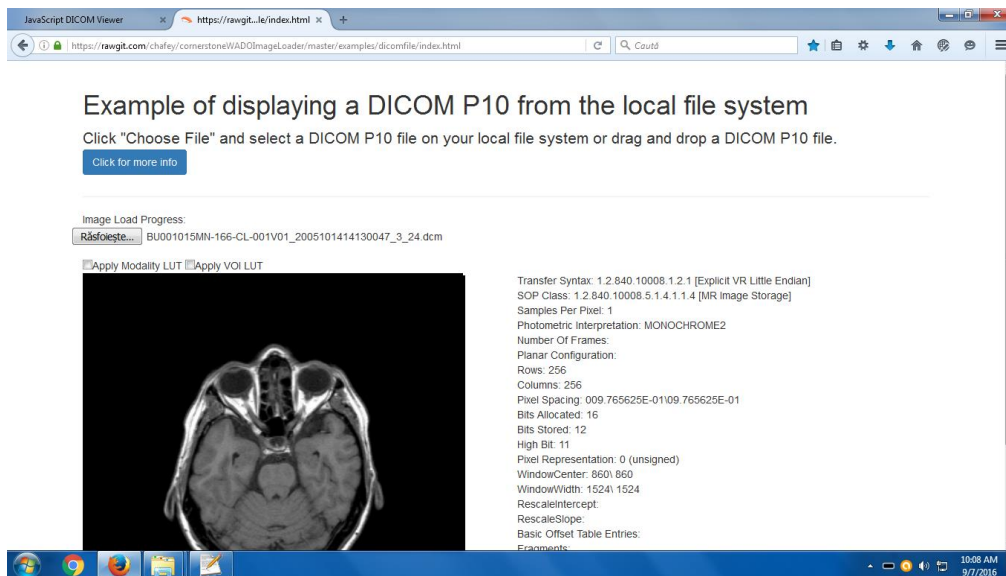


Figure 4.1. Cornerstone example application that displays a DICOM file from the local file system

(source:  
<https://rawgit.com/chafey/cornerstoneWADOImageLoader/master/examples/dicomfile/index.html>)

a more complex system has been developed that has additional options such as:

- image segmentation with a variable threshold;
- colorization;
- saving of individual slices on the local file system in PNG format;
- multiplanar 3D reconstruction based on the saved slices;
- loading of a set of slices saved from a previous session;
- interaction with the displayed 3D model either by the mouse or with a LEAP Motion controller.

Figure 4.2 shows the key functional modules of the resulting JSDV application.

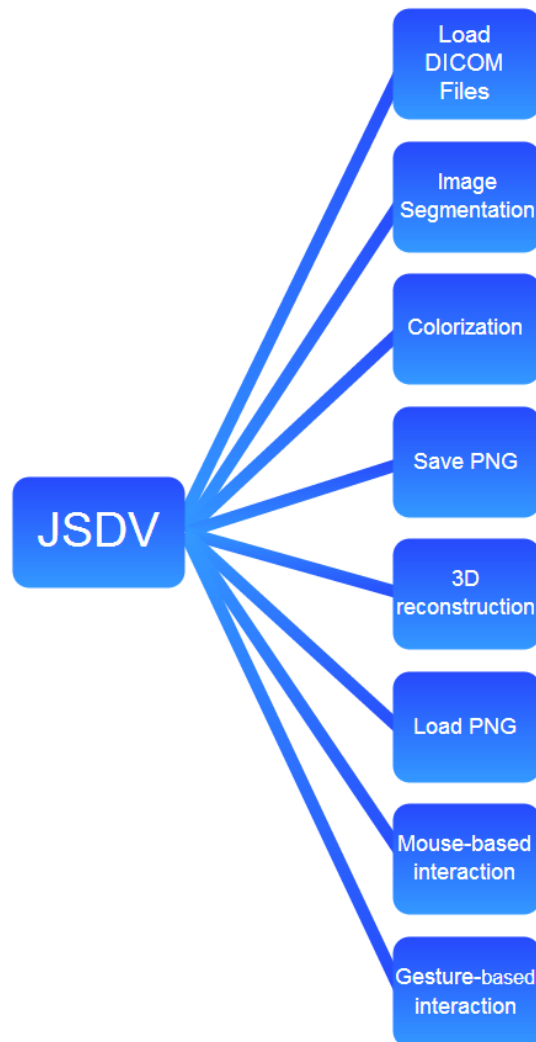


Figure 4.2. JSDV key functional modules

The application would not be possible without the advanced functionalities provided by HTML5, WebGL and LeapJS that will be further detailed throughout this chapter.

## 4.1. Application architecture and software modules

The resulting web application has a very simple and intuitive interface as shown in Figure 4.3.

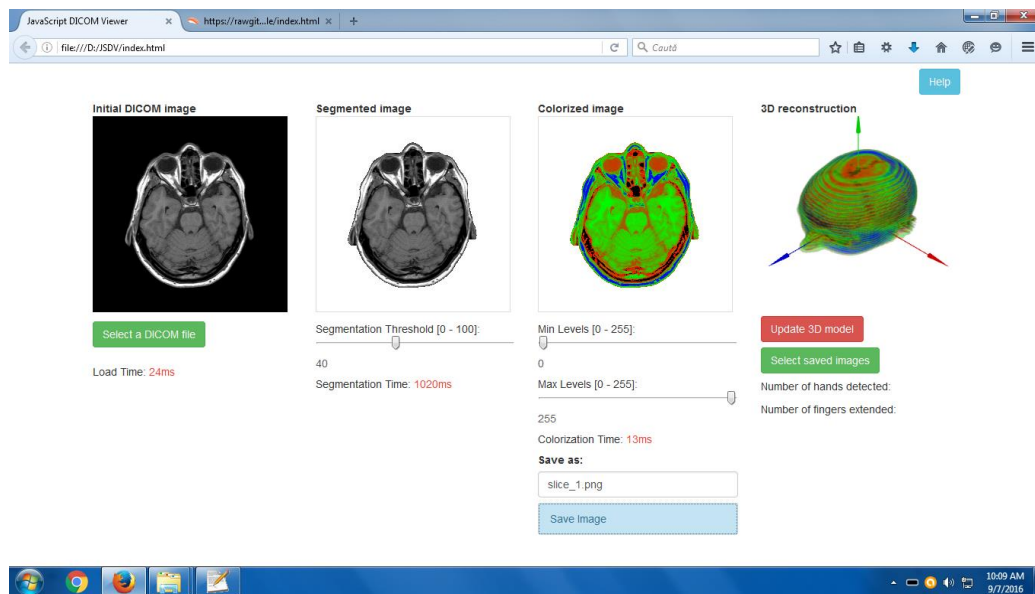


Figure 4.3. JSDV user interface

The application contains the following folders and source files that are well commented in order to give future developers more insight regarding the various solutions that have been implemented:

- *css* – the folder that contains the style sheets of the application;
- *img* – the folder used to store the PNG images extracted from the DICOM files;
- *js* – the folder that contains the various JavaScript files;
- *index.html* – the HTML file that is the main entry point of the application and can be opened in Firefox (or in Chrome with the following command line attributes `--disable-web-security --user-data-dir` , to disable the Same Origin Policy).

The diagram in Figure 4.4 is a visual representation of the libraries and web technologies that were used by the author of the thesis during the development of the JSDV application.

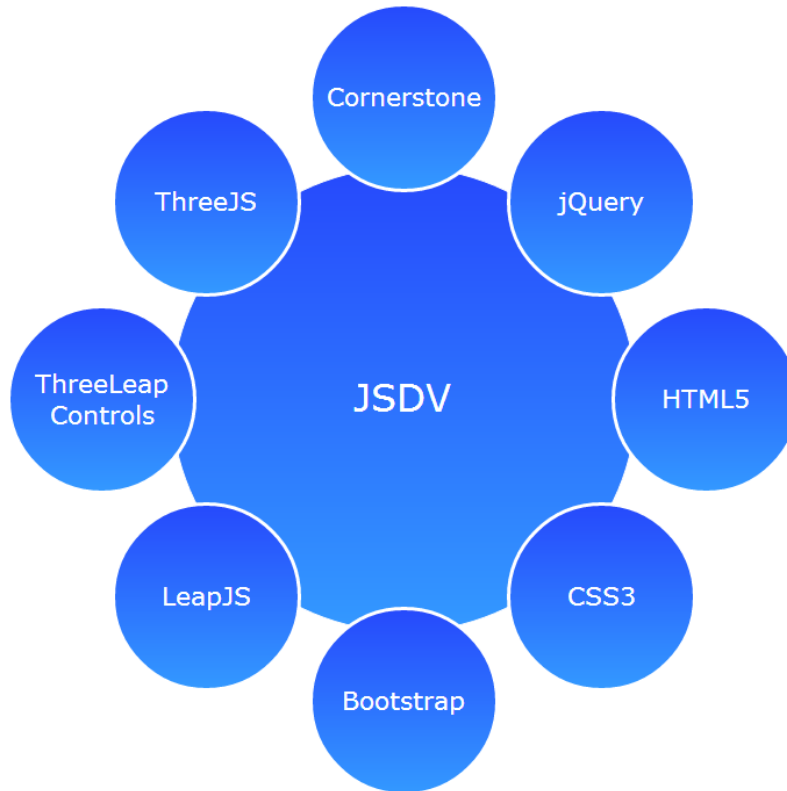


Figure 4.4. Web technologies used by JSDV

JSDV is written in HTML5 and uses JavaScript in order to interact with the various page elements making it entirely open source and paving the way for further development. It's worth mentioning that even if the application was designed to handle everything locally, the `cornerstoneWADOImageLoader` library is built to load remote DICOM files, retrieved via a HTTP GET request from a WADO (Web Access to DICOM Objects) server.

The current application is just an example of the complex interactions that can be achieved during the development of a browser-based medical visualization system using the following open-source libraries:

- *jQuery* – is a JavaScript library that simplifies the syntax needed to find and manipulate DOM elements. Since not everybody is familiar with this framework and because JavaScript in some cases is faster, the main code in JSDV is written in JavaScript.
- *Bootstrap* – a web user interface development framework that provides a mobile first approach for building responsive web pages and applications. I've added a series of CSS classes in the `style.css` file that can be uncommented in order to colorize the various elements of the page, as a visual aid during the user interface development.
- *Cornerstone* – a set of JavaScript libraries that allow the parsing of DICOM files, loading the extracted images and tools to interact with

them. It also has some other dependencies described in the source code (e.g. jQuery and ThreeJS).

- *ThreeJS* – a JavaScript library that greatly reduces the size and amount of the code that need to be written by software application developers when working with WebGL. This component has the role of rendering the reconstructed 3D model based on the studied patient's DICOM slices.
- *LeapJS* – a library that is used to track the positions of users hands and fingers with a LEAP Motion Controller facilitating the gesture-based interaction inside a web application.
- *ThreeLeapControls* – that allows the interaction between LeapJS and ThreeJS, by mapping the position of the hands and fingers detected by LEAP Motion to a 3D position in the rendered virtual scene. The library has two files: one is LeapObjectControls.js (that allows the interaction with a 3D object) and LeapCameraControls.js (for moving the camera in the rendered 3D scene).

Figure 4.5 illustrates the interaction between the user and the various modules of the JSDV application.

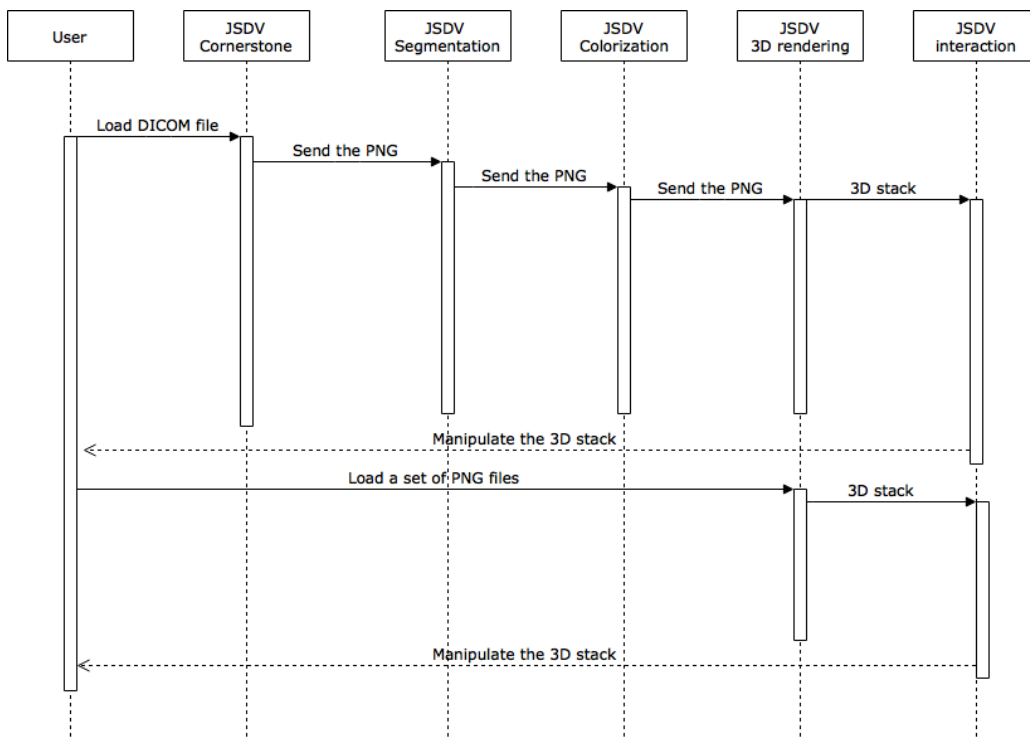


Figure 4.5. Interaction between the user and the various modules of JSDV

As shown in Figure 4.5 the steps required to obtain the 3D model and manipulate it (by gestures or a 2D mouse) are the following:

1. The user loads a DICOM file through the application user interface to extract the imagistic data.
2. The extracted PNG file is sent to the segmentation module.
3. The segmented image is sent to the colorization module.
4. The colorized image is used as a 2D plane for the multiplanar 3D reconstruction (3D rendering).
5. The resulting 3D stack is ready to be manipulated by the user.
6. The same functionality can be obtained by using a set of previously obtained PNG images.

This architecture allows switching between users in case that a LEAP Motion controller is used for the manipulation of the 3D stack, without having to stop the application for a separate training phase as it would be necessary if cameras would be used to track the hand motion.

The diagram in Figure 4.6 presents a use case for interacting with the rendered 3D model using a 2D mouse or a LEAP Motion device.

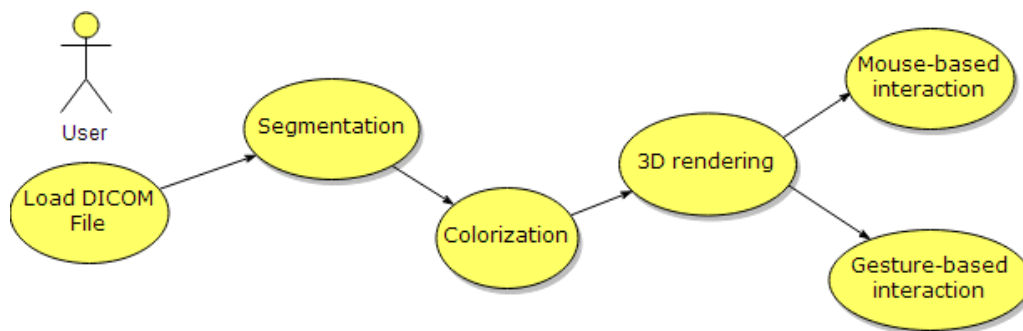


Figure 4.6. JSDV use case for interacting with the 3D model

The core JavaScript functions of JSDV used for extracting the pixel data from the DICOM files, segmentation, colorization, saving of the resulting images on the local file system, rendering the 3D model and manipulating with it with gestures are added to the *main.js* file and will be discussed in the following.

## 4.2. Image segmentation algorithm for restrictive environments

The image segmentation module is used to isolate the relevant pixels inside the image extracted from the DICOM file that is loaded using the "Select a DICOM file" button in the user interface. As we can see in Figure 4.7 the segmented image on the right has all the surrounding pixels turned into transparent pixels and the only visible area is the image of the patient's bones and soft tissues.

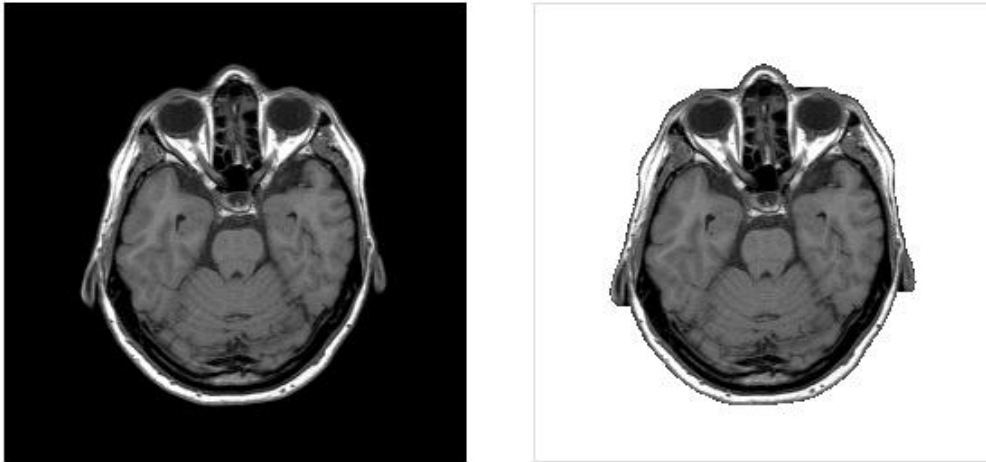


Figure 4.7. The initial DICOM image on the left and the segmented image on the right

We will use this image later as textures for multiple 2D parallel planes that will form the rendered 3D model of the investigated anatomical area.

The segmentation algorithm uses two passes through each pixel of the initial image:

- one to establish if a certain pixel belongs to the ROI (region of interest) marking it with 1, or not in which case we mark it with 0 inside an array that has the size of the canvas width and height matching the number of pixel rows and columns that needs to be analyzed.
- the second pass reads the associated values for each pixel and assigns a specific grayscale value to the corresponding pixels on the segmentation canvas. If the pixel belongs to the ROI it will copy the initial value and if not it will assign the value 0 to all the four RGBA components, rendering it to transparent black.

The threshold is used in the first pass to establish if the red component of a certain pixel is below a certain level (in this case is out of the ROI) and in the second pass to set the pixel colors of the segmented image and to verify if a certain pixel is below the threshold, but inside the ROI. Without this condition set the black pixels inside the ROI would be also set to transparent. The diagram presented in Figure 4.8 summarizes the segmentation process as described by the author of the thesis in [98] and [99].

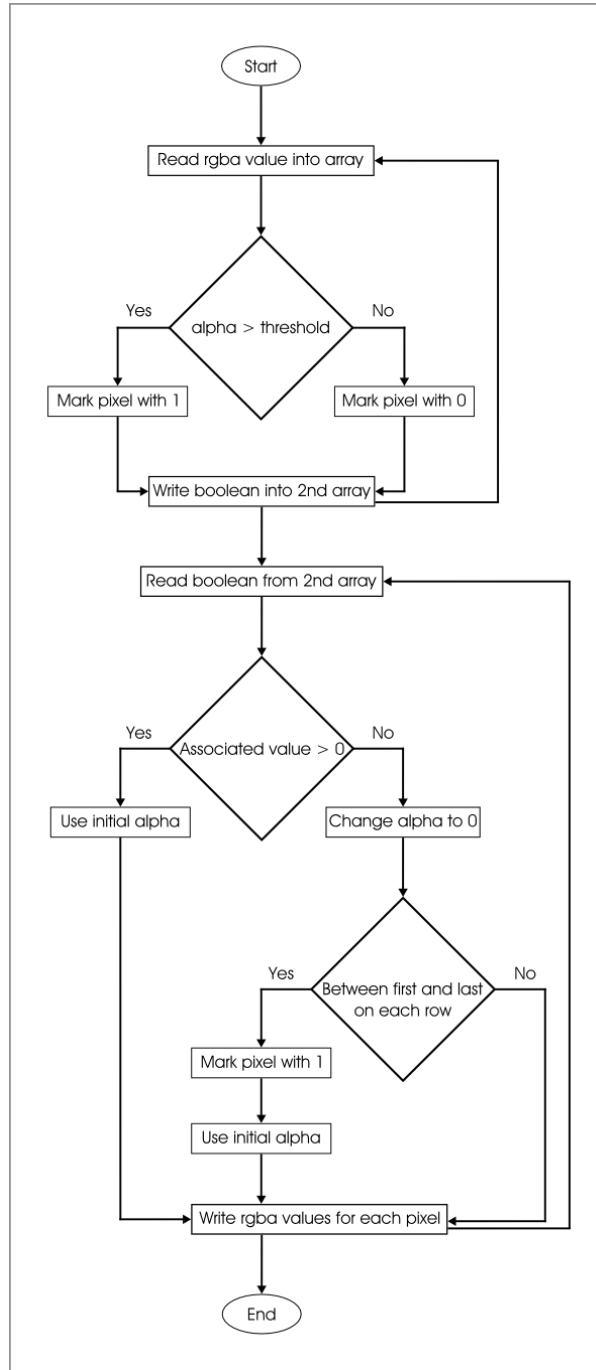


Figure 4.8. The segmentation process diagram presented by the author in [99]



By setting the segmentation threshold we can obtain an optimum image as shown in Figure 4.7. If we set it too low (or too high) the resulting image would have various artifacts that would render it unpractical for the 3D reconstruction as in Figure 4.9.



Figure 4.9. The segmentation threshold set to zero

The application uses HTML5 *canvas* elements and its associated canvas API that allows us to get each element's context and to retrieve each pixel's RGBA components that are stored in the *ImageData* object for the initial image. After that, the *putImageData()* method is used to copy the individual pixels in the segmentation context or we can set each of the four RGBA components separately by assigning the value 0 to all the four components through the *data* property of the *ImageData* object (making it transparent black).

### 4.3. Image colorization

The image colorization module is used to assign different RGBA color values to each pixel in the segmented image based on the tissue density as described by the author of the thesis in [100]. This feature provides a more detailed view to the physician during the image interpretation, as we can see in Figure 4.10 some of the pixels in the grayscale image on the left might have similar values that might be hard to distinguish between them.

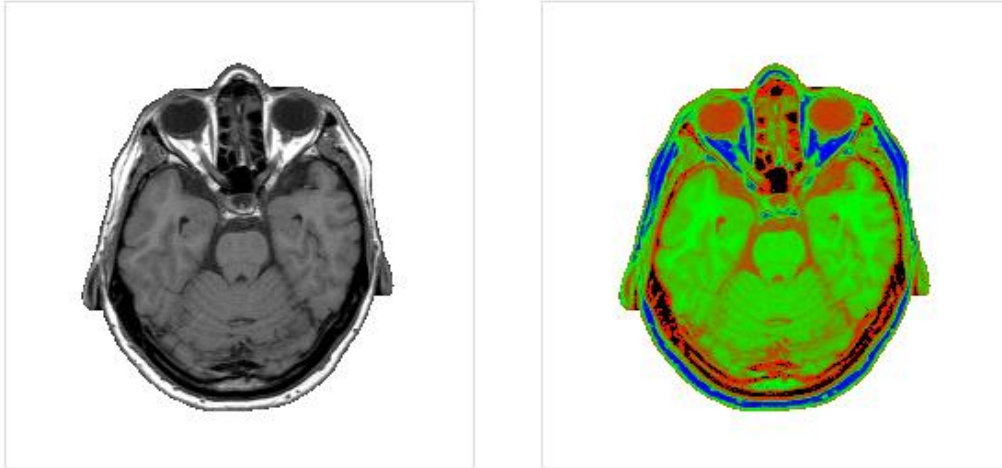


Figure 4.10. The segmented image on the left and the colorized image on the right

The algorithm used to colorize the image is based on Marisol Martinez Escobar's algorithm [101], adapted to the current situation when all the relevant tissues needs to be colorized (not only a ROI).

The imagistic data stored in the initial DICOM files is expressed in HU (Hounsfield Units) This is a scale to map the radiodensity of various tissues when exposed to electromagnetic radiation (such as X-rays used in CT). The HU values are transformed to grayscale levels in a process called windowing (done by cornerstone), because the HU values are usually between -1000 (for air in the lungs) and +3000 (for bone) depending on the CT vendor. The Table 5.1 present the most common HU values associated with various tissue types.

Table 5.1 The approximate HU values of various tissue types

Tissue type	HU value
air	-1000
lung	-900
fat	-100
water	0
white matter	+20
kidney	+40
grey matter	+45
spleen	+55
liver	+60
blood	+65
bone	+400 and above

In order to obtain the HU range two *input* elements has been added to the user interface that enables to set the ranges between 0 and 255. These *input* elements have attached two event listeners for the *onchange* event that trigger the colorization function. Every pixel that has its red component out of this range is set

to transparent black, this way we can isolate specific tissues inside the image as in Figure 4.11.

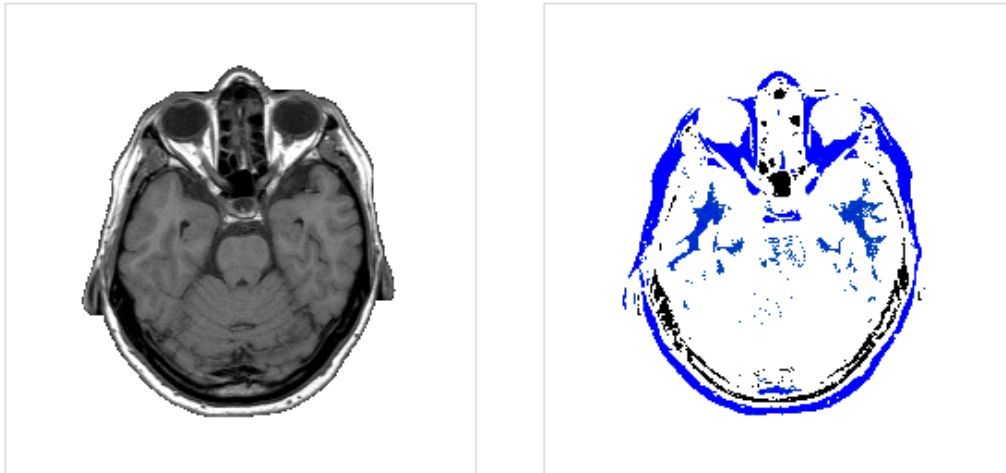


Figure 4.11. The bone tissue is isolated and colorized in the image on the right

The other pixels that have the red component inside the range are colorized based on the algorithm mentioned above by calculating a  $p$  parameter as the division between the current pixel's red component and the HU range. Based on the value of the  $p$  parameter, the corresponding  $RGB$  values are calculated for each pixel using the following formulas:

- if  $p$  is less than 0.5 the components are
 
$$\text{Red} = (1 - p * 2) * 255$$

$$\text{Green} = 255 * p * 2$$

$$\text{Blue} = 0$$
- if  $p$  is greater than 0.5 the components are
 
$$\text{Red} = 0$$

$$\text{Green} = 255 * [1 - (p - 0.5) * 2]$$

$$\text{Blue} = 255 * (p - 0.5) * 2$$

The  $A$  (*alpha*) component is set to a maximum of 255 and each pixel is colorized using the `putImageData()` method mentioned above.

The diagram presented in Figure 4.12 summarizes the image colorization process.

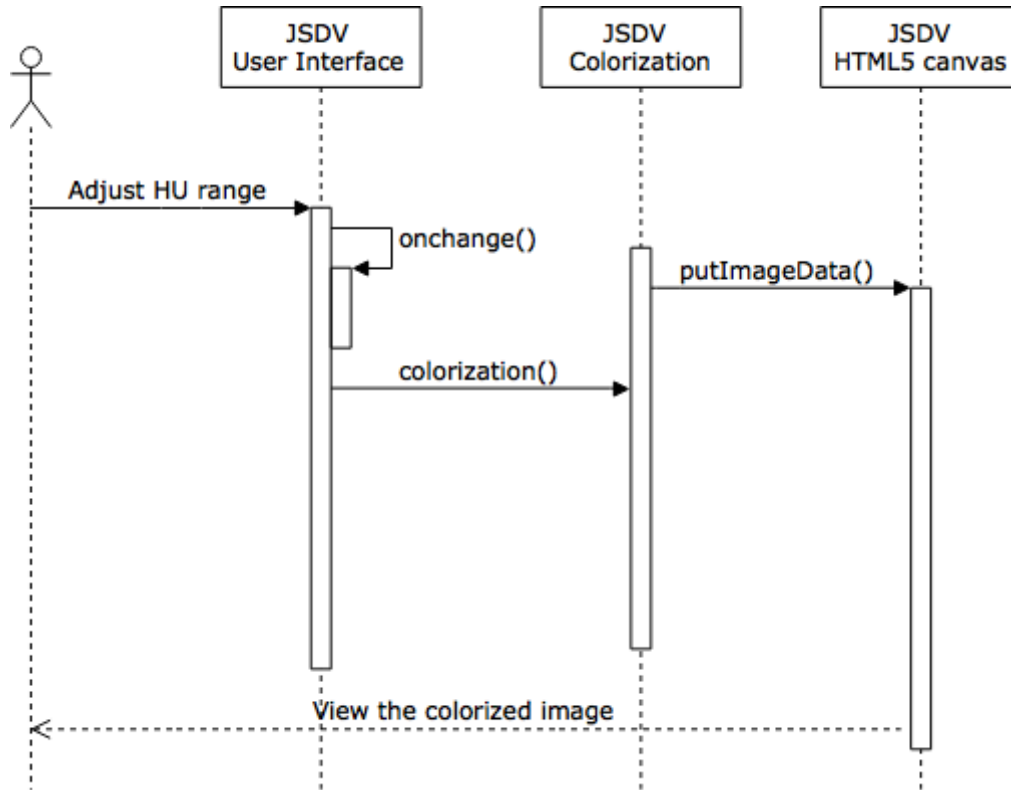


Figure 4.12. JSDV image colorization process

As the diagram shows the colorization takes place in real time as the user adjusts the *input* elements on the JSDV application's user interface preparing the images for the 3D reconstruction.

#### 4.4. Multiplanar 3D reconstruction

The 3D reconstruction has the role to provide a more complex view than the classic study of the individual 2D slices with a negatoscope. The application uses the ThreeJS library to render a 3D scene where the 3D model, camera and lights can be added. In Figure 4.13 is presented a 3D model that is rendered based on the previously colorized slices.

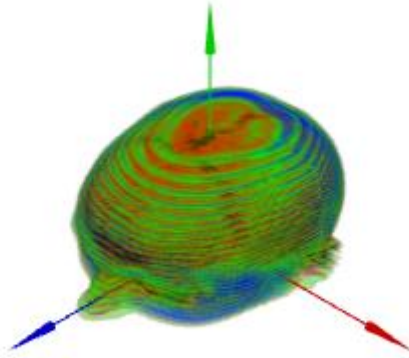


Figure 4.13. A 3D model rendered based on the colorized slices

After the segmentation and colorization phases the processed images can be saved as individual PNG files on the local file system (by using the `cornerstoneTools.saveAs()` function) and the “Update 3D model” button will be shown (which is initially hidden by having the `collapse` class from Bootstrap). This button will listen for an attached `onclick` event and when the event is triggered it will call the `updateRenderer()` function in its callback in order to start loading the images.

The 3D scene is setup in the `initRenderer()` function that is called by the jQuery `ready()` function when the DOM tree is fully loaded. The `initRenderer()` has two parameters (`canvasWidth` and `canvasHeight`) that we will transmit in order to set the size of the renderer and the camera aspect ratio because we will render the 3D model only on the assigned canvas, not the entire window. If the browser supports WebGL (detected by the Detector.js library) a new `THREE.WebGLRenderer` object is created that enables hardware-accelerated rendering by accessing the GPU directly. The `THREE.WebGLRenderer` object will provide a canvas with a “webgl” context, so there is no need to set this separately. Further away we will set the renderer size and we will use the `setClearColor` property to set the background color of the canvas to white and opaque (with the parameters `0xffffff` and `1`). After we setup the renderer, we append its DOM element (the resulting canvas) to the div with the id `stackedImages` that will serve as its parent element. This step is done using jQuery with the following single line of code: `$("#stackedImages").append(renderer.domElement);`

Further in the `initRenderer()` function we will create a new camera by using a `THREE.PerspectiveCamera` object. The properties of this newly created object will determine what we see in the 3D scene (the viewing frustum) as follows:

- `fov` – sets the camera frustum vertical field of view;
- `aspect` – sets the aspect ratio, usually the window width divided by window height, in our case is the ratio between the horizontal and vertical size of the canvas;
- `near` – the camera frustum near plane (the minimum distance from the camera from where the rendered scene becomes visible, if this is set to

a larger value we will need to provide a mechanism to scroll in on the z axis towards the scene to make it visible);

- *far* – the camera frustum far plane determines the maximum distance from the camera that we can see.

The camera's XYZ position is set with the *camera.position* properties and after that we point the camera towards the center of the scene by setting its *camera.lookAt* property.

After setting up the camera we will create a new *THREE.LeapCameraControls* object that allows us to control the camera position with a LEAP Motion controller.

Next we will need a scene where the 3D model will be added later. This is achieved by creating a new *THREE.Scene* object. The *scene.add()* method is used to add all the other objects to the newly created scene (the new *ArrowHelper* and *PointLight* objects).

The last thing that we will create in the *initRenderer()* function will be a *THREE.Object3D* object that we will use in *updateRenderer()* to add each slice for the multiplanar reconstruction.

In the Figure 4.14 a diagram is presented that shows the various ThreeJS objects used for the multiplanar 3D reconstruction as described previously.

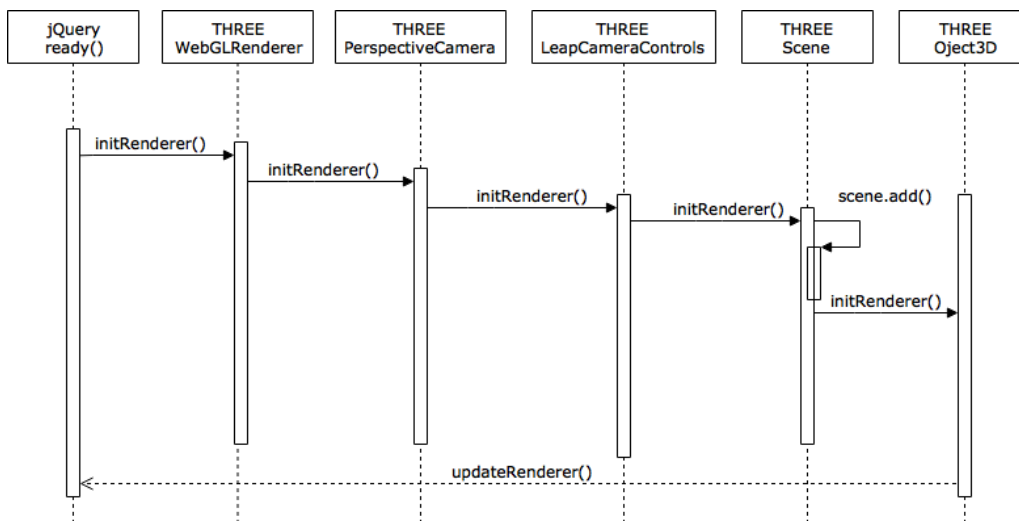


Figure 4.14. JSDV multiplanar 3D reconstruction

As shown in the diagram the 3D reconstruction uses several objects and methods provided by the ThreeJS library such as:

- THREE.WebGLRenderer;
- THREE.PerspectiveCamera;
- THREE.LeapCameraControls;
- THREE.Scene;
- THREE.Object3D.

These objects are making it easier to developers to build a complex 3D scene.

#### 4.5. Loading the 2D slices

The application has two modalities to load the saved 2D slices: one after the images are saved during the processing of each DICOM file using the “Update 3D model” button and the second using the “Select saved images” pseudo-button (that’s actually an HTML *input* element that is made to look like a button) to load the PNG files stored in the “img” folder of the application in a previous session. Both options will call the *updateRenderer()* function in their event listener callback. In the following we will discuss the second option in more detail because it has a more complicated mechanism.

First of all in order to have a better user experience the author used a simple hack to style the *input* element to make it look like a button. This is possible by wrapping the *input* element in a *label* that has the *btn-file* class provided by Bootstrap, and setting the *input* element’s *display* property to *none*. The difference between the two is major as we can see in Figure 4.15.

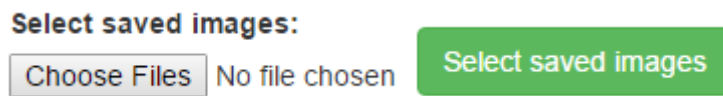


Figure 4.15. Difference between the initial input element on the left and the pseudo-button on the right

This *input* element of type *file* will listen for an attached *onchange* event and when the event it’s triggered it will execute the code inside its callback function. This function sets the *sources* array length property to 0 in order to empty the array (in case we previously used it). We will use this array in the *updateRenderer()* function to retrieve each opened files name. After that we will retrieve all the *File* objects from the *FileList* object. The *FileList* object is returned by the *files* property of the *input* element. In our example we retrieve it in the callback from the event’s *target* property and store it in the *files* variable (like this *var files = e.target.files;*). The contents of the files are retrieved in a for loop using a *FileReader* object, that allows the tracking of files as they are loaded (with the *FileReader.onload* event handler) and to read the contents of each file (with the *FileReader.readAsText()* method). After that we have access to the name of the current file that is loaded and can add it to the *sources* array. Finally it will call the *updateRenderer()* function that starts loading the images as 2D textures.

The *updateRenderer()* function has the role to load each selected PNG file as 2D textures to a set of stacked 2D planes used for the 3D multiplanar reconstruction as in Figure 4.16.

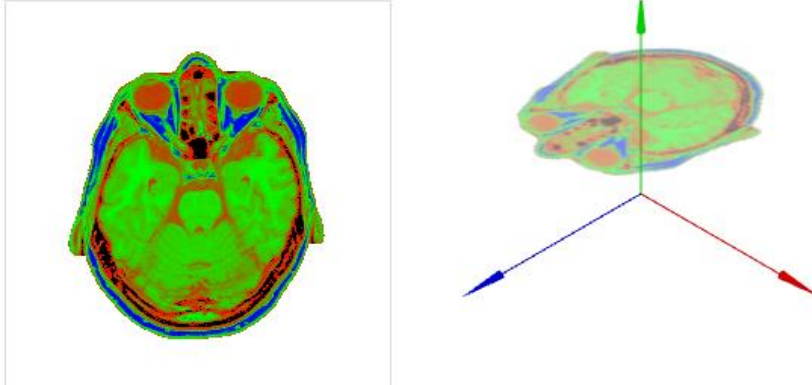


Figure 4.16. A 2D plane from the 3D stack that uses as texture the image on the left

This module uses a *THREE.PlaneGeometry* object in order to create a 2D plane and a *THREE.MeshBasicMaterial* object to create a material that will be used as properties of a *THREE.Mesh* object in order to create a mesh (a set of vertices, edges and faces). The *THREE.MeshBasicMaterial* object will have two properties: the first is *map* that will map the loaded PNG file as a texture and a second called *side* that we will set as *THREE.DoubleSide* in order to make the texture visible on both sides of the plane (the *side* property is inherited from the *THREE.Material* object). The PNG files are loaded using the *THREE.TextureLoader* object's *load* event handler (in our code `textureOfSlices = textureLoader.load('img/' + loadedFileName);`). The *THREE.LoadingManager* object is used to track the number of images loaded by the *THREE.TextureLoader* and to add the *scansCube* to the scene (`scansCube` is the *THREE.Object3D* object created in `initRenderer()`). The *THREE.LoadingManager* object's *onLoad* handler will also call the `render()` function in its callback and will render the final scene and camera. The sequence is described in Figure 4.17.

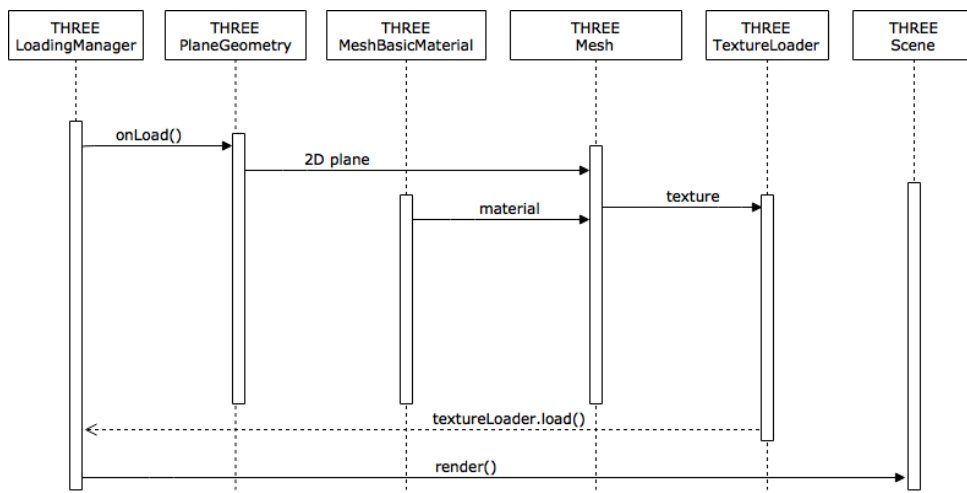


Figure 4.17. The process of loading images as textures



While the images are loaded a set of new *THREE.LeanObjectControls* objects will be created for each slice. These objects will allow the user to control each individual slice with a LEAP Motion controller making it possible to rotate, scale or translate them across the rendered scene. This functionality provides an enhanced visual experience.

#### 4.6. Controlling the 3D model with a LEAP Motion Controller

Besides listening for mouse events, the JSDV application has an alternative way to control the displayed 3D model when a LEAP Motion controller is detected. This device allows the tracking of the 3D position and movements of the user's hands and fingers that are in its field of view with a great accuracy.

In order to use a LEAP Motion controller we need to install the LEAP Motion SDK that will provide a local WebSocket server (running as a service or a daemon) and a separate LeapJS library that will process the tracking data retrieved from the server as a JSON-formatted message. Because LEAP Motion provides 200 frames of tracking data per second and a browser usually repaints the screen at 60 frames per second we need to use a *Leap.loop()* function to process the data sent by the controller only when the browser is ready to draw. The *Leap.loop()* function (provided by the Leap Motion API ) will set up a Leap *controller()* object and WebSocket connection. These can be specified by the user in the options parameter of the *Leap.loop()* function or we can use the default options. The second parameter of the *Leap.loop()* is a callback function that will be called internally using the browser's *requestAnimationFrame()* method. This way the latest frame of Leap Motion data will be passed to the application whenever the browser is ready to update the onscreen animation. In JSDV we will use the default options and the code looks like this: *Leap.loop(function(frame) {...});* .

From the *Frame* object we can retrieve the number of hands detected with the following line of code *var numberOfHandsDetected = frame.hands.length;*. To access the first hand detected we will use *frame.hands[0]* and in each hand object we will retrieve the first finger using the following code *hand.fingers[0]* . To check if the first finger detected is extended or not we can use the following code *hand.fingers[0].extended* .

In the JSDV application we used *ThreeLeapControls*, a library written by Torsten Sprenger, that allows the mapping of the position of the hands and fingers detected by LEAP Motion to a 3D position inside the scene rendered by the *ThreeJS* library. As previously mentioned the *ThreeLeapControls* library comes with two options: one is *LeapObjectControls.js* (that allows the interaction with a 3D object) and *LeapCameraControls.js* (for moving the camera in the rendered 3D scene).

Because *ThreeLeapControls* uses LeapJS v 0.2.0 and *ThreeJS* rev. 59, the author of the thesis needed to rewrite some parts of the code in order to make it compatible with the current LeapJS v 0.6.4 and *ThreeJS* rev. 79. The resulting code is used to detect the number of fingers that are extended and also for distinguishing each individual finger. This makes it possible to associate specific fingers with specific actions such as rotation, scaling or translating the rendered 3D objects.

Inside the *Leap.loop()* callback function the application calls a *showCursor()* function that is used to set the *top* and *left* CSS properties of a *div* element with the id *cursor* based on the number of hands and fingers extended. By doing this a red

dot (the *cursor* div) will be shown on the canvas when the user points the index finger towards a specific slice as shown in Figure 4.18.

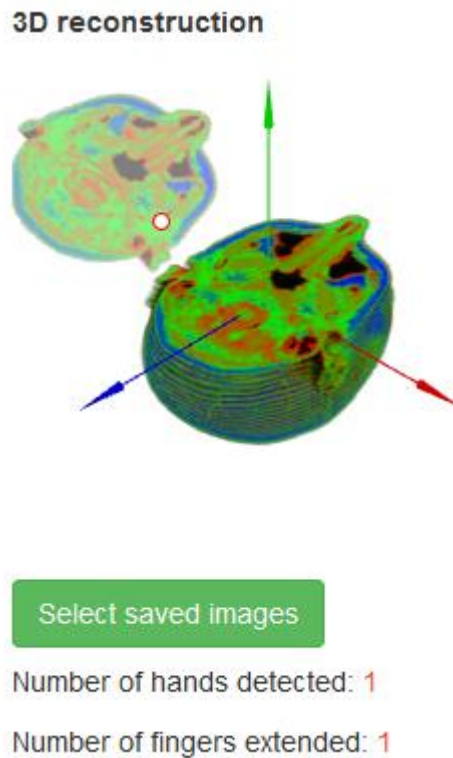


Figure 4.18. The cursor is shown when one finger and one hand is detected

The *showCursor()* function is used only to have a visual feedback regarding the position of the cursor and the slice that is currently selected. The actual mechanism that calls the update function of either an instance of a *LeapObjectControls* object or a *LeapCameraControls* object is based on the value returned from the *focusObject()* function (called in the *Leap.loop()* callback). If the value returned is -1, the *cameraControls* object will be updated on each frame, otherwise the current object inside the *objectsControls* array will be updated (we have a separate *LeapObjectControls* object for each slice).

In order to have a visual feedback regarding the slice that is currently selected a *changeControlsIndex()* function is called every 200 ms by a *setInterval()* method inside the jQuery *ready()* function. This will set the opacity of the slices based on the value returned by the *focusObject()* function stored in the *controlsIndex* variable as shown in Figure 4.19.

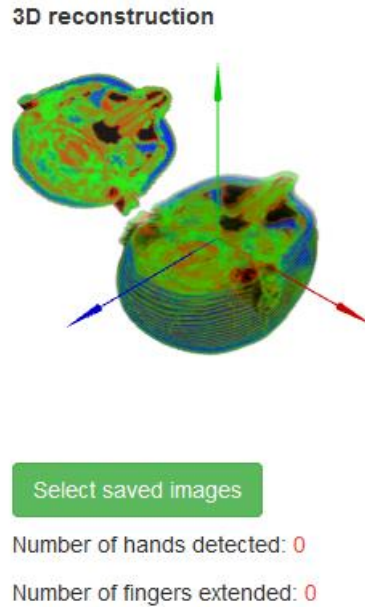


Figure 4.19. The opacity of the selected slice changes based on the number of hands and fingers detected

The actual rotation, scaling or panning of the selected object is done by calling the *update()* function of either an instance of a *LeapObjectControls* object or a *LeapCameraControls* object as previously mentioned. The sequence of controlling the 3D model with a LEAP Motion controller is illustrated in Figure 4.20.

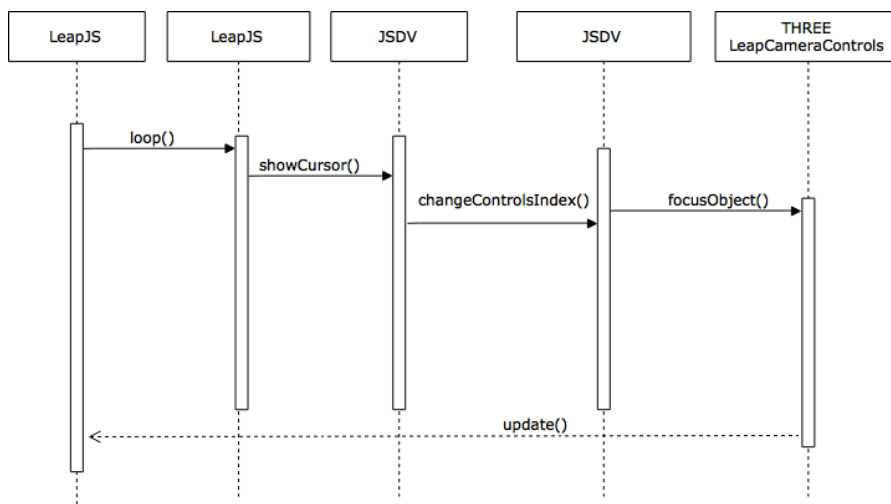


Figure 4.20. The sequence of controlling the 3D model with gestures

The *update()* function will call its internal functions that will deal with the transformations needed in order to move a 3D object (or the camera) based on the number of hands and extended fingers that are detected.

## 4.7. Conclusions

The current chapter detailed the development phases and results of the JSDV web application based on the open source software technologies presented in Chapter 3. As demonstrated, the users can interact with the displayed 3D images using simple gestures providing an enhanced user experience during image interpretation. The application can be easily adapted for use in conjunction with a pseudo-holographic device by rendering the 3D image in a modal window (available in Bootstrap). The browser-based approach allows the deployment of the code on any operating system without the need to install a virtual machine or any third-party browser plug-ins.

The advantages of the simplified segmentation algorithm developed by the author of the thesis are derived from the fact that it has to check only for a single neighboring pixel resulting in increased script execution times, a key factor that needs to be considered during the development of any browser-based application. Even if this approach may result in a rough contour of the obtained edges, as the author described it in the current chapter, this shortcoming might be surpassed by using a variable threshold as it is implemented in the application.

The new segmentation algorithm is not limited only to static images and might be used in other areas where the fast extracting of specific features from images might be essential (e.g. surveillance, automotive and aerospace industries). The execution speed of an application that analyzes video frames rather than specific images would grow considerably since instead of verifying up to eight additional pixels to establish if a specific pixel belongs to the ROI, the current algorithm has to check for a single one.

Other open source applications such as OsiriX [102] or VolView [103] are also available for visualization of the patient's DICOM files, however these applications depend on a previously installed operating system that in some situations might limit their portability (e.g. OsiriX runs only on Mac OS X) and impose other financial restrictions (e.g. due to the operating system's proprietary license). Neither one of them is implementing gesture-based interaction nor provides an interface suitable for visualization on a pseudo-holographic device.

The JSDV application has been tested during the research by a large amount of volunteers and the results will be presented in Chapter 6.

## 5. INTERACTING WITH A 3D MOLECULAR VIEWER USING GESTURES

The present chapter is meant to underline the versatility offered by any open source oriented application architecture. In order to demonstrate this, the author of the thesis extended the GLmol web application by adding extra functionalities for the user interaction mechanism and presented it in [104]. The newly implemented gesture-based manipulation features are possible due to the openly available LeapJS library that can be interfaced with GLmol using ThreeLeapControls. This way the application works both with a classical mouse and with a LEAP Motion Controller if available, making it easier to the users to manipulate the displayed 3D molecular structures.

### 5.1. The GImol 3D molecular viewer

GLmol [17] is a client-side 3D molecular viewer developed by Takanori Nakane, which facilitates the visualization of chemical structures stored in openly available databases (e.g. RCSB PDB, PubChem) under various file formats. The current version of the application is 0.47 and it was developed in 2012 using HTML5 and JavaScript for the main interaction code, while the 3D models are rendered in any browser that has WebGL support by using a customized version of the ThreeJS library (version 49). GLmol has a dual open-source license of LGPL3 and MIT license. The application offers a simple browser-based user interface as we can see in Figure 5.1.

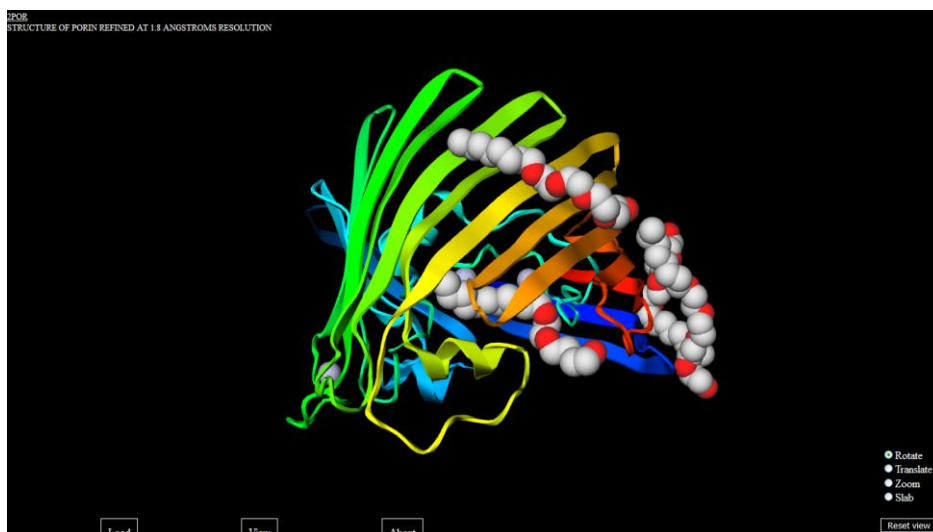


Figure 5.1. GLmol user interface

## 86 5. Interacting with a 3D Molecular Viewer Using Gestures

The application user interface has three tabs: one for loading the chemical structures from various remote or local sources, one for display options and another that briefly explains how to use it. On the right side there are also some radio buttons to change the mouse functionality when interacting with the application such as: rotate, translate, zoom and slab.

The *View* tab offers numerous display options that can change the representation of the displayed 3D chemical models as shown in Figure 5.2 and also the possibility to take a screenshot of the web page that can be saved on the local file system as a 2D raster image for later use in presentations.

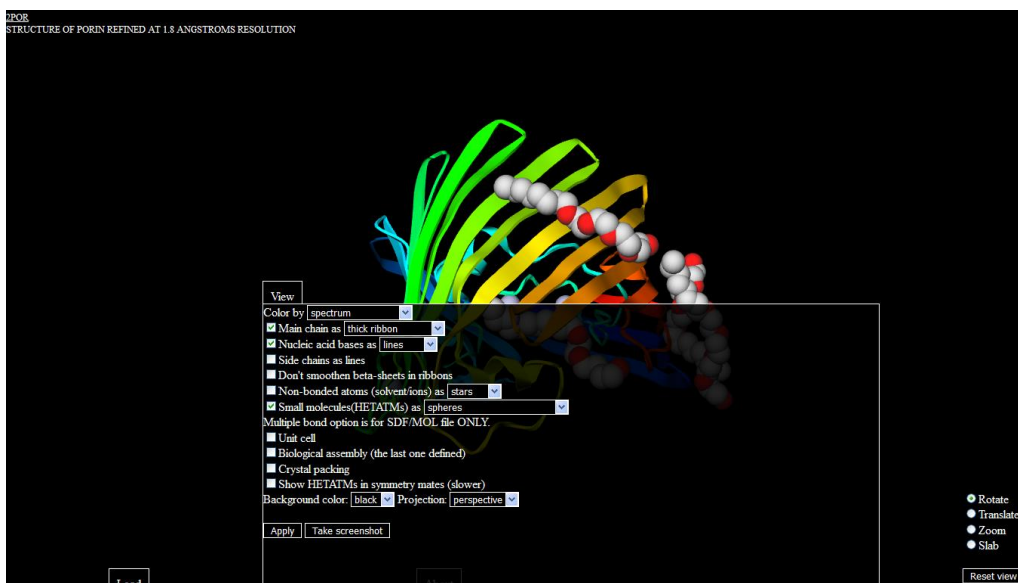


Figure 5.2. GLmol display options

GLmol despite the fact that it has been developed as a browser-based application offers a large number functionalities such as:

- loading local PDB (Protein Data Bank), SDF (Structure Data Format) or XYZ file formats (using the *File API* provided in HTML5);
- loading remote PDB files from the RCSB (Research Collaboratory for Structural Bioinformatics) PDB server;
- loading remote SDF files from the NCBI (National Center for Biotechnology Information) PubChem server;
- various colorization options such as: by spectrum, chain, secondary structure, B factor, polar/nonpolar;
- main chain representation as: thick ribbon, thin ribbon, strand, cylinder and plate, C alpha trace, B factor Tube and bonds;
- nucleic acid bases representations as: sticks, lines or polygons;
- side chains representation as lines;
- smoothing of beta-sheets in ribbons;
- non-bonded atoms representation as spheres or stars;

- small molecules representation as: sticks, ball and stick (also with multiple bonds), spheres, icosahedrons and lines;
- display unit cell;
- show crystal packing;
- show HETATMs in symmetry mates;
- changing the background color (white, black or gray);
- changing the projection (ortographic or perspective);
- rotating, panning and scaling the displayed 3D model with the mouse;

The rendered 3D chemical models can be directly embedded inside web pages resulting in a more interactive experience that is very suitable for virtual laboratories.

The diagram in Figure 5.3 shows an interaction example between the user and the Glmol application.

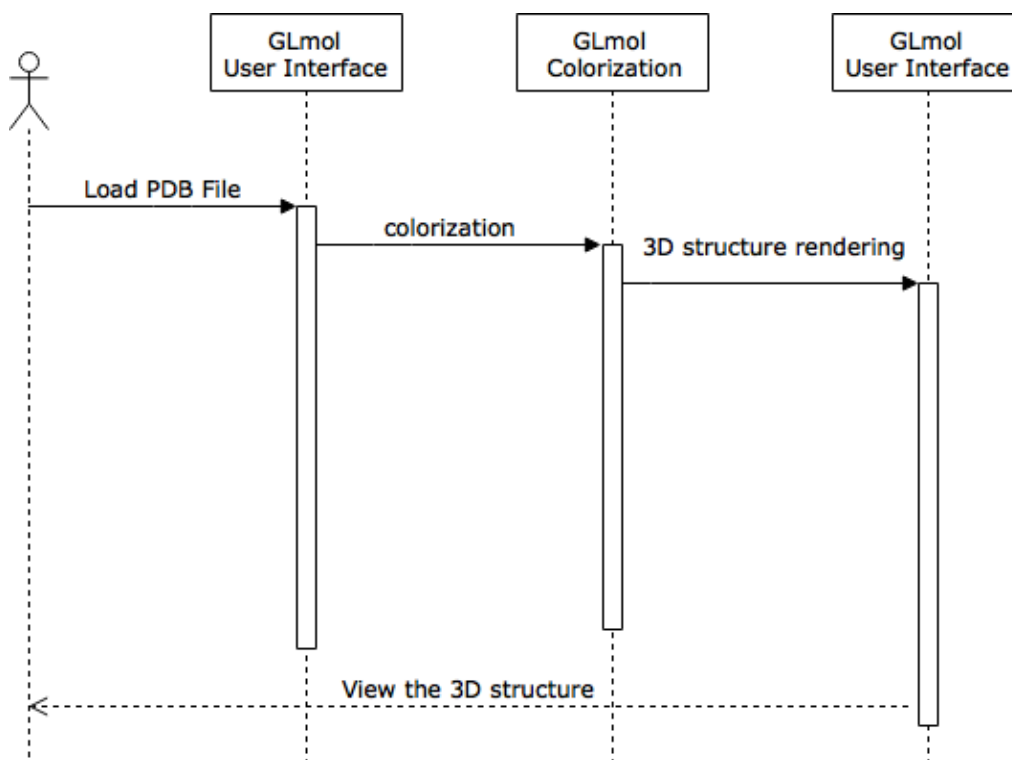


Figure 5.3. Interaction between the user and the Glmol application

After the 3D chemical structure is rendered by the Glmol application the user can rotate, pan and scale it with a 2D mouse. As previously shown by the author of the thesis this interaction method can be improved by adding gesture-tracking functionalities to the application.

Figure 5.4 provides a simplified overview of the main functionalities of the initial Glmol application (rendered in blue) and the additional functionalities added by the author of the thesis (rendered in orange).

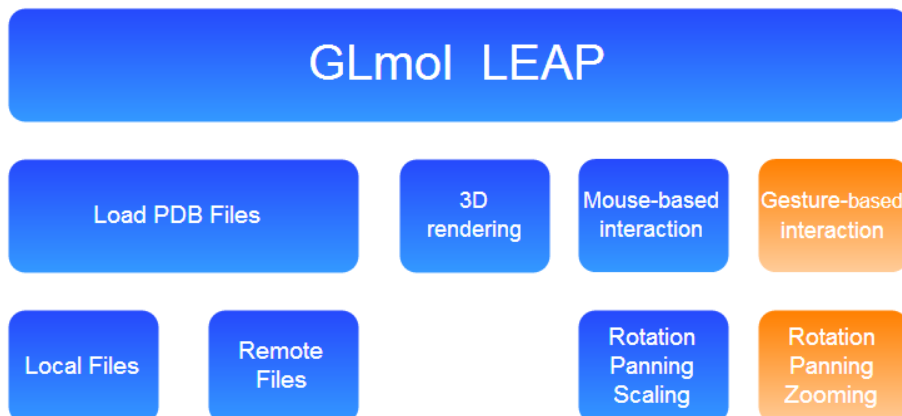


Figure 5.4. GLmol LEAP main functionalities

As the diagram in Figure 5.4 shows the main contribution is the addition of the gesture-based functionalities, which also implied the modification of the initial code of GLmol that will be detailed in the following.

## 5.2. Adapting the application to gesture-based interaction

The initial GLmol application in the 0.47 version uses jQuery version 1.7 and a custom built version of the 49<sup>th</sup> revision of ThreeJS. In order to implement the gesture-based interaction the author of the thesis used LeapJS version 1.4.4 and LeapObjectControls that had to be modified to work with the functionalities provided by the custom ThreeJS version used by GLmol. To achieve this, the *sub()* method had to be replaced with the older *subSelf()* and the *add()* method with the *addSelf()* method provided in the earlier ThreeJS revision.

The diagram in Figure 5.5 provides an insight regarding the initial web technologies used by GLmol (rendered in blue) and the additional libraries (rendered in orange) added by the author of the thesis in order to achieve the gesture-based functionalities after extending the application.





Figure 5.5. Web technologies used for GLmol LEAP

GLmol also had initially the JavaScript application entry code embedded in the html file that has been separated by the author of the thesis in a separate *main.js* file for a better maintainability.

The application starts with a new instance of a *GLmol* object that is assigned to a *glmol01* variable inside the main application code. This new object calls all the subsequent methods in order to load and render the 3D model of the chemical files.

To achieve the gesture-based control of the displayed 3D models, the developers need to install the LEAP Motion SDK that will add a local WebSocket server (that runs as a service or a daemon) and to call the LeapJS library that processes the hands and fingers positions supplied by the controller as a blob of JSON data. After that the developer will have to add a *Leap.loop()* method (available in the Leap Motion API) inside the *defineRepFromController()* method of GLmol. *Leap.loop()* will assure that the data sent from the LEAP Motion Controller is processed only when the browser is ready to repaint the web page. This is a necessary step since the controller is capable to generate 200 frames per second, while the browser usually repaints the screen only at a rate of 60 frames per second.

In the *Leap.loop()* method the developers can use the default options with the callback function that updates the window by using the *frame* object with the following code: `Leap.loop(function(frame) {...});`. Inside this callback function we will call the *update* method (provided by LeapObjectControls) for every *frame* on a *modelControls* instance returned by the following code: `var modelControls = new THREE.LeapObjectControls(this.camera, this.modelGroup);`. This new

*LeapObjectControls* instance has two parameters provided by the *GLmol* object: one of the *camera* that is viewing the scene and one for the rendered 3D model (the *modelGroup* object). Further by calling the *glmol01.show()* method inside the *Leap.loop()* method's callback function, *GLmol* renders the 3D model by using a *THREE.WebGLRenderer*() instance.

The diagram in Figure 5.6 illustrates the interaction between the various modules of the *GLmol* LEAP application in order to render the 3D model.

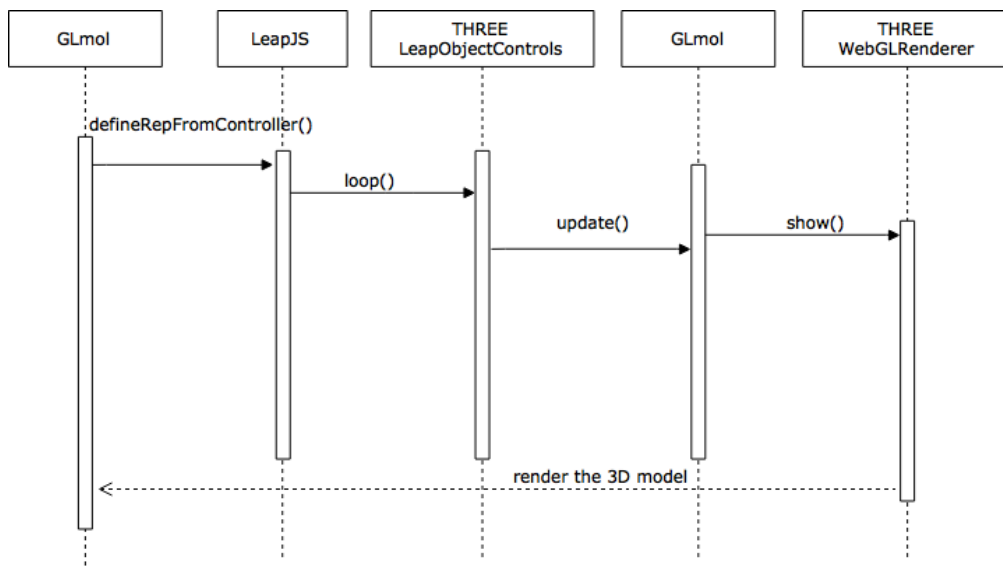


Figure 5.6. Interaction between the modules of the *GLmol* LEAP application

Similar to the *JSDV* application the developers can retrieve the number of hands detected from the *Frame* object using following line of code `var hds = frame.hands;` inside the *LeapObjectControls* instance. This allows associating the number of hands detected to various actions such as the rotation, zooming or panning of the displayed 3D models.

The advantage of adding these extensions to *GLmol* is that the application will enable the user interaction both with a classical mouse and a LEAP Motion device, facilitating the manipulation of the rendered 3D molecular structures that also can be visualized on a pseudo-holographic device as shown in Figure 5.7.



Figure 5.7. Testing the GLmol application on a HoloVision display

The diagram in Figure 5.8 presents a use case for interacting with the rendered 3D chemical structure using a conventional 2D mouse or a gesture-tracking device such as the LEAP Motion controller.

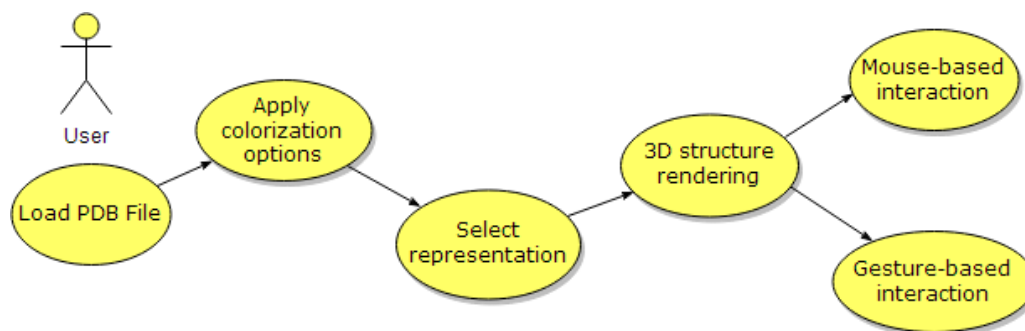


Figure 5.8. GLmol LEAP use case for interacting with the 3D model

The GLmol's user interface makes it very suitable to display the 3D content on an autostereoscopic 3D display since the application already provides an option to set the background color that needs to be black in order to achieve a better contrast on such devices.

The resulting extended version of GLmol was named GLmol LEAP by the author of the thesis and will be referenced accordingly in the next chapter that will present the results of testing the application by professionals involved in computational chemistry.

### **5.3. Conclusions**

As the current chapter demonstrates the previously described software technologies are a suitable solution to developers in any other areas where the users need to interact with complex 3D models. Similar areas might include architecture, aeronautical engineering, computer simulators.

The original contributions of the author of the thesis as described in the current chapter are the adaptation of the GLmol application to gesture-based interaction that adds a new dimension to the user interface. This approach is also meant to sustain the supposition that the open source architecture of an application allows the extension of it by adding extra functionalities that the initial author might not have had planned in the development phase.

The work in shown in Chapter 5 emphasizes the generalization possibilities of the models, algorithms and methods from Chapter 4 and the flexibility for implementing them in new domains.

## 6. EVALUATION

As the last part of the research, both JSDV and GLmol LEAP presented in chapters 4 and 5 have been tested on specific user groups. The author of the thesis started with separate test plans for each application. These test plans included the usability test objectives, methodology (equipments, participants' characteristics, and procedures), usability tasks description, usability metrics and usability goals as it should be organized according to [105]. For the test plans the author of the thesis used a template [106] provided by *Usability.gov* a site that is managed by the Digital Communications Division in the U.S. Department of Health and Human Services' (HHS) Office of the Assistant Secretary for Public Affairs. These templates serve as a blueprint for a test plan, are available in the public domain and have been adapted by the author of the thesis for both applications since they have a general character. Each test plan had 7 annexes that included the following:

- Annex 1 - a consent form that each participant completed before the actual testing where they confirmed that the participation is voluntary and that they agreed to participate in the test. For this annex the author of the thesis adapted the template provided by *Usability.gov* [107] for each application;
- Annex 2 - a pre-test demographic and background information questionnaire regarding age, gender, professional status, previous experience with the equipment and software used for testing;
- Annex 3 - a table containing the task scenarios (task description, what is required to perform by the user, succesful completion criteria and a maximum time for completion) used by the test moderator during the testing;
- Annex 4 - a time on task table used by the moderator to record the time for completion of each task completed by the participants;
- Annex 5 - a five-point Likert scale usability questionnaire inspired by the questions from the Software Usability Measurement Inventory (SUMI) [108] questionnaire adapted separately to each tested application. This questionnaire provides answers regarding specific quantitative metrics for each application;
- Annex 6 - a five-point Likert scale 2D mouse assessment of comfort questionnaire based on the ISO 9241-9 [109] and the presentation of the standard from [110]. This questionnaire provides answers regarding some subjective metrics regarding the ease of use of the mouse by the users related to the applications.
- Annex 7 - a five-point Likert scale Leap Motion device assessment of comfort questionnaire also based on the ISO 9241-9 and presentation mentioned above. This questionnaire provides answers regarding some subjective metrics regarding the ease of use of the LEAP Motion device by the users.

For both applications the author of the thesis used a custom-built pseudo-holographic 3D display (Figure 6.1), which allowed the user to have a more vivid representation of the displayed 3D model resulting in a better understanding of the

studied images. This autostereoscopic display was built by the author of the thesis from a cardboard box painted in black and a sheet of thin glass inserted at a 45 degree angle to reflect the images displayed by a regular 2D monitor placed on top of it. This solution represents a low-cost alternative to any contemporary autostereoscopic display, having the advantage of a reduced weight making it suitable for transportation if testing occurs in multiple locations. Unlike a HoloVision display that has around 80 kilograms this solution has around 600 grams and provides a practical option for explaining the concept and working principles of a 3D pseudo-holographic display.



Figure 6.1. Custom-made pseudo-holographic 3D display

The resulting images displayed by such a simplified version of an autostereoscopic display are not comparable with those provided by a HoloVision display where the rendered 3D object seems to float in front of the display due to the mirror used to focus the image, it rather represents an affordable alternative due to the costs of the materials used for its construction.

The rest of the chapter deals with detailing particular aspects and the methodology of testing for each application (regarding apparatus, participants, training and procedures) and the analysis of the results.

### **6.1. Usability evaluation overview for the JSDV application**

The scope of the usability testing for the JSDV application was to measure user performance and user-satisfaction levels that would help identify eventual problem areas within the applications' user interface.

The usability evaluation took place in a controlled testing environment with the help of a pool of representative users and included the testing of the following functionalities of the JSDV application:

- locating the applications' help menu;
- loading of a set of DICOM files of the skull area (publicly available samples were obtained from <http://www.microdicom.com/downloads.html>);
- testing of the interactive segmentation module;
- testing the application's colorization module;
- saving the individual slices as PNG files on the local file system;
- loading a set of previously saved PNG images;
- 2D mouse-based interaction with the rendered 3D model;
- gesture-based interaction with the displayed 3D model by using a LEAP Motion Controller (selection and separation of individual slices from the 3D stack);

The tasks needed to perform by the users were established by the author of the thesis for covering the most important functionalities provided by the JSDV application, considering the short time for which the participants were available (usually 15 minutes). The tasks were the same for all the involved participants and the task scenarios were the following:

- Find out how to start using the JSDV application;
- Locate and open a DICOM file for segmentation;
- Obtain a well defined contour of the image loaded from the DICOM file;
- Isolate a tissue type in the colorized image;
- If satisfied with the colorization result, save the resulting image for the 3D reconstruction;
- Load a set of images for the 3D reconstruction;
- Rotate the rendered 3D model with the mouse;
- Select and isolate a specific slice from the 3D stack with the help of the LEAP device;

The task scenarios, successful completion criteria, procedures required to perform and a maximum reference time for completion were described in annex 3 of the test plan. A separate time on task table (annex 4) was used to record user performance such as the time for completion per task for each user, as well as critical and non-critical errors for later processing. This provided answers regarding successful completion rates, error-free rates and time on task averages. The completion rates were calculated as the percentage of users that completed the task without critical errors. Error-free rates were calculated as the percentage of participants who had no errors at all (neither critical nor non-critical errors).

Three separate questionnaires (annex 5, 6 and 7) have been used to evaluate the usability metrics that provide insights regarding the ease of use of the application and user satisfaction when confronted with a LEAP Motion against a 2D mouse.

After the completion of the task scenarios described above, a five-point Likert scale usability questionnaire consisting of 14 questions (annex 5) was used to record answers from each participant regarding the following quantitative metrics:

- consistency;
- learnability;
- relevancy;

- clinical applicability;
- user control and freedom;
- software response speed;
- help section related issues;

The subjective metrics were evaluated with the help of two questionnaires (annex 6 and annex 7) based on the ISO 9241-9 assessment of comfort questionnaire. These were meant to provide answers regarding the participants satisfaction when using a 2D mouse or a LEAP Motion Controller to manipulate the rendered 3D model.

## 6.2. Methodology for testing the JSDV application

The user group that participated in the usability test consisted of 24 medical students from the Victor Babes University of Medicine and Pharmacy Timisoara and 8 master students from the Faculty of Mechanical Engineering of the Politehnica University of Timisoara (specializing in Implants, Prostheses and Biomechanics Assessment). The testing took place in the B019 lab from the Automation and Applied Informatics Department of the Politehnica University of Timisoara.

For the testing the author of the thesis used a Lenovo ideapad Y510P laptop, that had an Intel Core i7-4700MQ CPU at 2.40 GHz, 8GB RAM, an Nvidia GeForce GT755M graphics card and a display resolution of 1920x1080 at 60 Hz, together with a custom built autostereoscopic 3D display in conjunction with an Asus monitor at a resolution of 1920 x 1080 pixels and a LEAP Motion Controller. The laptop had a Windows 10 Education 64-bit operating system, Google Chrome v 56.0.2924.87, Mozilla Firefox v 51.0.1 browsers and LEAP Motion software ver. 2.3.1. Figure 6.2 shows the entire setup used to test the 3D interface of the JSDV application.

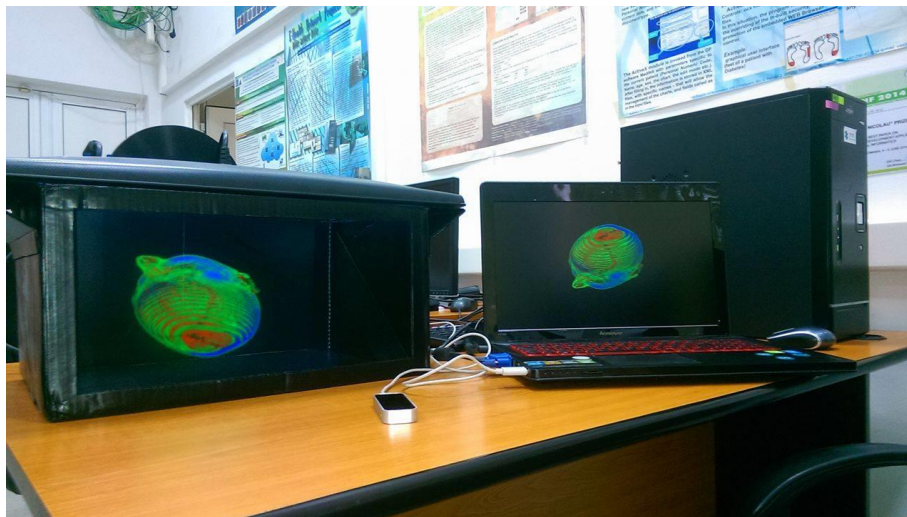


Figure 6.2. Setup used for testing the 3D user interface of the JSDV application



Before the test each participant completed a consent form regarding their willingness to participate (annex 1) and a demographic and background information questionnaire (annex 2).

After the test the participants also completed a usability questionnaire (annex 5) and two separate device assessment of comfort questionnaires for the 2D mouse version (annex 6) and for the LEAP Motion (annex 7).

A detailed description regarding the participants, training and specific procedures will be provided in the following.

### 6.2.1. Participants

The JSDV application has been tested by 32 volunteers (10 male, 22 female). The participants' ages ranged between 18 and 26 year old, with an average of 20.28 years old, distributed as in Figure 6.3.

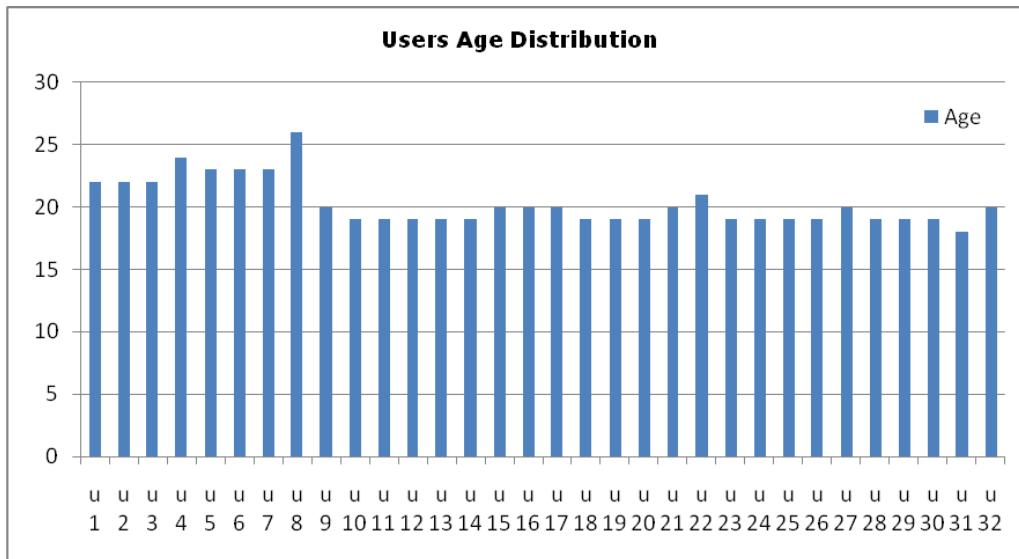


Figure 6.3. JSDV users age distribution

Their educational background ranged from 24 medical students to 8 master students (some of them being also medical students or physicians). Only three participants had a previous clinical experience (from 3 months to 4 years). Three participants had a specialization in medical rehabilitation and one in medical engineering.

All participants declared that they had previous experience using a computer (between 6 and 16 years). Only three participants declared that they had a previous experience with medical software (from one to four years). Four participants declared that they had previous experience with 2D medical datasets (between 3 and 5 years) and only two of them had experience with 3D medical datasets (1 and 2 years). All participants declared that they are comfortable using a web browser. While 31 of the participants declared that they are comfortable using a 2D mouse, only 2 of them had a previous experience with a LEAP Motion Controller. All

participants were right handed and 3 of them declared that they had a previous experience with a 3D pseudo-holographic display.

### 6.2.2. Training

The participants had a brief overview of the usability test procedure and were subjected to a ten minutes training regarding the application and equipment used. The participants were divided into smaller groups and each group had a separate training session to make sure that everybody can see the user interface details and understands the role of each device used for testing the application as shown in Figure 6.4.



Figure 6.4. A small group of participants during the training phase

After answering the eventual questions regarding the equipment and the application the volunteers were asked to complete the task scenarios without any further aid from the test observer.

### 6.2.3. Procedure

In order to have a controlled environment and to obtain consistent results all the usability testing sessions took place in the B019 lab from the Automation and Applied Informatics Department of the Politehnica University of Timisoara. The participants were divided into smaller groups due to the estimated time of 15 minutes for each volunteer needed to complete the questionnaires and the test scenarios. The participant's interaction with the JSDV application was monitored by the author of the thesis (who assumed the role of the moderator) seated next to the users during evaluation.

The author of the thesis instructed the participants regarding the procedures, apparatus used and the JSDV application, emphasizing that they are evaluating the application, rather than the moderator evaluating them in order to alleviate any stress factors that might have influenced the test results. After that each participant signed a consent form (annex 1) confirming that their participation is voluntary and that they might raise any concerns during the test session. The volunteers also completed a pre-test demographic and background information questionnaire (annex 2) regarding their age, gender, professional status, clinical experience, specialty, previous experience with computers, medical software, 2D or 3D medical datasets, web browser usage, handedness and the equipment used for testing (2D mouse, LEAP Motion Device, Pseudo-holographic 3D display). The author of the thesis explained the participants that he will measure the amount of time taken to complete each task scenario and that they should not have any other exploratory behavior outside the ones required to complete the task excepting the fact that they were allowed and encouraged to use the application's help section whenever they felt the need to do so. After that the moderator read out loud each task (from annex 3) and instructed the volunteers to start. The tasks were considered finished either if the participant indicated so, either if they exceeded the previously estimated time for completion.

The times for completion, critical and non-critical errors were registered in a table (annex 4) for post processing. The non-critical errors were recorded in cases when the participant completed by himself the task without any kind of guidance from the moderator, but did not find the needed functionalities right away. The critical errors were the result of not being able to complete the task or exceeding the estimated time for completion by the participants. The critical errors were recorded as the reference time plus one second (e.g. for 120 it was recorded as 121) due to the fact that the participants were stopped by the test moderator if they exceeded the time.

After the completion of the tasks the participants were asked to complete the post-test usability questionnaire (annex 5) and the device assessment of comfort questionnaire for the 2D mouse (annex 6) and for the LEAP Motion Controller (annex 7).

### 6.3. Test results for the JSDV application

The evaluation results for the JSDV application showed that the participants were able to complete the tasks on average under the previously estimated time (used as reference). In some cases the results were beyond the expectations resulting in major differences (such as in case of tasks 3, 4 and 7). Table 6.1 presents the reference times for each task, mean (average), the minimum, maximum times for completion and the population (N = 32) standard deviation (SD).

Table 6.1 JSDV time on task results showing the reference, average, minimum, maximum times and standard deviation

Task no.	T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8
----------	-----	-----	-----	-----	-----	-----	-----	-----

<b>Reference time (s)</b>	120	60	120	120	60	60	60	120
<b>Average (s)</b>	38.69	21.72	7.41	5.53	15.06	24.75	2.47	26.59
<b>Min time (s)</b>	8	3	1	1	1	7	1	5
<b>Max time (s)</b>	121	61	80	58	45	61	8	121
<b>Population SD (s)</b>	28.61	18.20	13.45	9.72	10.11	15.18	1.39	34.17

Even if the average times for completion were different, the t-test results showed no conclusive differences between the times recorded for each task in case of the medical students ( $n = 24$ ) and the master students ( $n = 8$ ). Table 6.2 shows the average times for completion, unbiased (corrected) sample variances and sample standard deviations for both user groups as well as the p-values obtained for a two-tailed test where the two samples had unequal variance. Bessel's correction had been used to calculate the sample variances ( $n-1$  was used instead of  $n$  in the calculations). The medical students are noted with MED and the master students with MSc in the table.

Table 6.2 JSDV average times on task, corrected sample variances, sample standard deviations and t-test results for the two user groups

<b>Task no.</b>	<b>T 1</b>	<b>T 2</b>	<b>T 3</b>	<b>T 4</b>	<b>T 5</b>	<b>T 6</b>	<b>T 7</b>	<b>T 8</b>
<b>Reference time (s)</b>	120	60	120	120	60	60	60	120
<b>Average MSc (s)</b>	55.13	32.25	15.75	11.13	11.25	42.75	1.75	62.75
<b>Corrected sample variance MSc</b>	1894.98	467.36	685.07	365.27	95.07	417.93	0.50	2707.64
<b>Sample SD MSc (s)</b>	43.53	21.62	26.17	19.11	9.75	20.44	0.71	52.04
<b>Average MED (s)</b>	33.21	18.21	4.63	3.67	16.33	18.75	2.71	14.54

<b>Corrected sample variance MED</b>	436.69	266.95	10.77	5.88	106.58	43.24	2.30	193.91
<b>Sample SD MED (s)</b>	20.90	16.34	3.28	2.43	10.32	6.58	1.52	13.93
<b>p-values</b>	0.207	0.124	0.269	0.307	0.231	0.013	0.023	0.035

The author of the thesis concluded that the differences between the averages of the two user groups were produced by the 10 critical errors recorded in case of the master students. The main reason for these critical errors was that the participants from the master students group tested the application in the late evening, after a long day of work and were tired, not being able to focus during the training session. As shown in Table 6.2 this affected the averages for the tasks 1, 2, 6 and 8 where the critical errors had been recorded. In order to solve the detected problems the JSDV user interface has been changed by adding a contrasting border to the *Select a DICOM file* button and by changing it back to its default value only after the user clicked the pseudo-button using its attached *onchange* event described in chapter 4.

Table 6.3 shows the recorded times per task scenario for the entire population. The non-critical errors are marked with bolded text and the critical errors with italicized text.

Table 6.3 Recorded times per task scenarios for JSDV

<b>Task no.</b>	<b>T 1</b>	<b>T 2</b>	<b>T 3</b>	<b>T 4</b>	<b>T 5</b>	<b>T 6</b>	<b>T 7</b>	<b>T 8</b>
<b>Reference time (s)</b>	120	60	120	120	60	60	60	120
<b>User 1</b>	32	44	7	58	2	61	3	<b>50</b>
<b>User 2</b>	11	3	<b>80</b>	4	18	61	2	<i>121</i>
<b>User 3</b>	<b>90</b>	61	5	9	15	61	2	<i>121</i>
<b>User 4</b>	<i>121</i>	11	12	4	5	29	1	61
<b>User 5</b>	<b>28</b>	20	3	2	3	61	1	7
<b>User 6</b>	38	61	2	1	27	35	2	15
<b>User 7</b>	<b>106</b>	<b>28</b>	10	4	19	14	2	<i>121</i>
<b>User 8</b>	15	30	7	7	1	20	1	6

<b>User 9</b>	<b>79</b>	<b>29</b>	5	4	4	20	2	18
<b>User 10</b>	<b>50</b>	12	6	6	18	9	2	31
<b>User 11</b>	<b>98</b>	46	2	2	3	16	4	8
<b>User 12</b>	11	3	2	3	2	27	3	7
<b>User 13</b>	33	<b>59</b>	3	3	<b>28</b>	18	4	8
<b>User 14</b>	39	28	6	14	21	22	3	<b>44</b>
<b>User 15</b>	14	7	3	3	5	23	4	<b>38</b>
<b>User 16</b>	43	5	4	4	12	10	1	7
<b>User 17</b>	18	28	4	3	27	25	1	58
<b>User 18</b>	11	<b>12</b>	1	3	<b>45</b>	9	1	8
<b>User 19</b>	<b>45</b>	<b>52</b>	2	2	<b>26</b>	10	2	14
<b>User 20</b>	8	7	3	4	4	7	3	14
<b>User 21</b>	<b>30</b>	<b>42</b>	8	4	18	<b>31</b>	3	8
<b>User 22</b>	25	12	3	4	9	15	2	7
<b>User 23</b>	<b>39</b>	11	14	3	7	18	4	5
<b>User 24</b>	<b>33</b>	5	4	2	20	17	3	14
<b>User 25</b>	41	8	13	2	11	25	2	11
<b>User 26</b>	36	8	4	3	12	24	1	6
<b>User 27</b>	26	4	3	3	18	20	3	8
<b>User 28</b>	31	12	8	5	22	<b>29</b>	3	7
<b>User 29</b>	20	16	2	2	12	18	1	5
<b>User 30</b>	25	8	5	3	25	<b>23</b>	2	7
<b>User 31</b>	31	7	2	2	26	16	3	11
<b>User 32</b>	11	16	4	4	17	18	<b>8</b>	5

The completion rates and error-free rates were calculated for all the tasks to determine the user performance based on the time on task (time for completion for each task).

The completion rate shows the percentage of participants that completed the task with no critical errors. Error-free rates are showing the percentage of users who had no errors at all during the completion of the task. Table 6.4 shows the separate completion rates (CR) and error-free rates (EFR) for the entire population, for the master students (noted with MSc) and for the medical students (noted with

MED). The last column shows the average completion rates and error-free rates for all the tasks.

Table 6.4 Completion rates and error-free rates for JSDV

<b>Task no.</b>	<b>T 1</b>	<b>T 2</b>	<b>T 3</b>	<b>T 4</b>	<b>T 5</b>	<b>T 6</b>	<b>T 7</b>	<b>T 8</b>	<b>Average</b>
<b>CR %</b>	96.88	93.75	100.00	100.00	100.00	87.50	100.00	90.63	<b>96.09</b>
<b>EFR %</b>	65.63	75.00	96.88	100.00	90.63	78.13	96.88	81.25	<b>85.55</b>
<b>MSc CR %</b>	87.50	75.00	100.00	100.00	100.00	50.00	100.00	62.50	84.38
<b>MSc EFR %</b>	50.00	62.50	87.50	100.00	100.00	50.00	100.00	50.00	75.00
<b>MED CR %</b>	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
<b>MED EFR %</b>	70.83	79.17	100.00	100.00	87.50	87.50	95.83	91.67	89.06

The high averages for the entire population (N = 32) of all the tasks regarding completion rates (CR = 96.09 %) and error-free rates (EFR = 85.55 %) are showing how well the users performed when first confronted with the JSDV application. These averages are high for the group of the master students (n = 8) and even higher in the case of the medical students (n = 24) where the average completion rate was 100.00 % and the average error-free rate was 89.06 %.

The chart in Figure 6.5 shows the different average completion rates for the entire population, for the master students group (n = 8) and for the medical students group (n = 24).

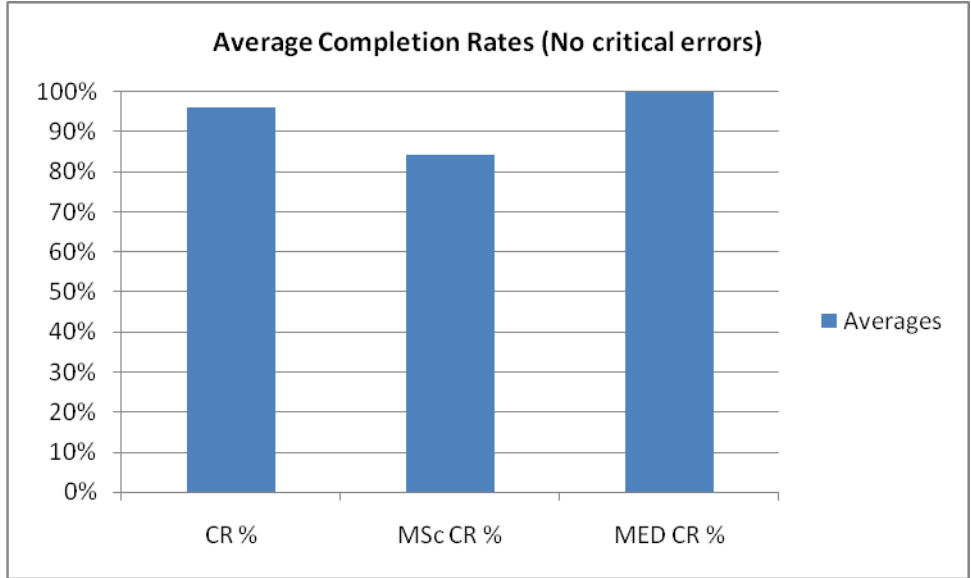


Figure 6.5. JSDV average completion rates

The chart also shows that even in the worst case there were only a few critical errors when the users didn't complete specific tasks by themselves.

The chart in Figure 6.6 provides an insight regarding the different error-free rates regarding the entire population (N = 32), for the master students (n = 8) and for the medical students (n = 24).

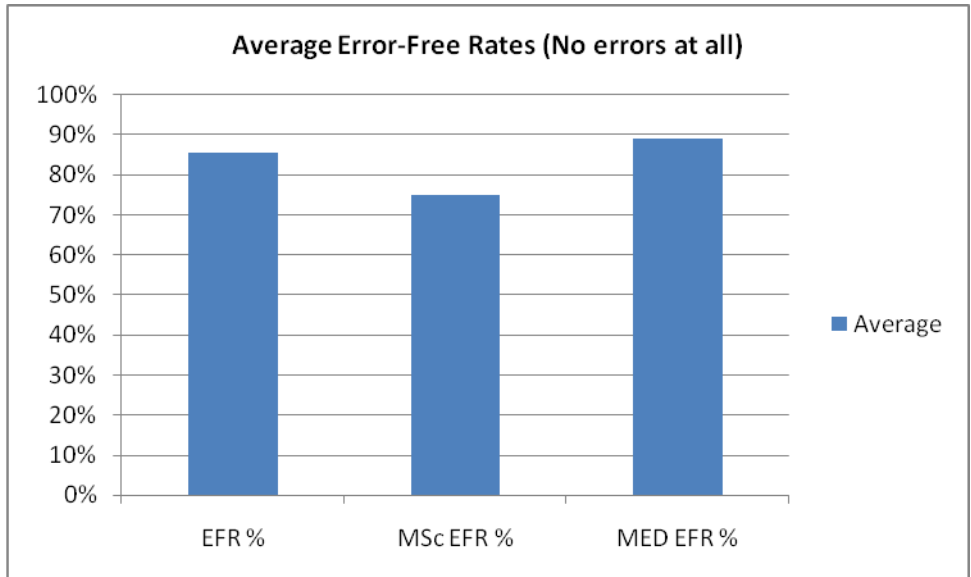


Figure 6.6. JSDV average error-free rates



The chart covers all the errors (critical and non-critical) for all the evaluated task scenarios.

The high average completion rates (96.09 %) and error-free rates (85.55 %) for all the tasks considering the entire population (N = 32) showed that the JSDV application overall was a success from the point of view of the user performance taking into consideration that this is the first time that the users interacted with it and that the vast majority of them haven't used a LEAP Motion device before.

The usability metrics were evaluated using the annex 5 where the participants expressed their opinions regarding the following affirmations:

1. I feel comfortable using this software.
2. The information presented on the screen is clear enough to understand how to operate the JSDV application.
3. I think the software responds too slowly at certain stages.
4. I have enough information to operate the application.
5. I think the user interface is organized in a simple and consistent manner.
6. Learning to operate the software was too hard.
7. I think that the help section is not very useful.
8. I would like to use this software at my workplace.
9. It was easy to find out what to do next when I got stuck.
10. The response time of the software is really fast.
11. There is no relevant information on the screen in order to operate the software.
12. There is too much information that I have to understand before I can operate the application.
13. The way the buttons are organized is logical and relevant.
14. I would recommend this software to my colleagues.

The annex 5 was inspired by the questions from [108] and adapted to the current situation.

The possible responses were on a five-point Likert scale (ranging between YES, yes, uncertain, no and NO). The following usability aspects were evaluated:

- user control and freedom (noted in the following with UC) - questions 1 and 4;
- consistency (noted in the following with C) - questions 2 and 5;
- software response speed (noted in the following with SR) - questions 3 and 10;
- learnability (noted in the following with L) - questions 6 and 12;
- help section related issues (noted in the following with HS) - questions 7 and 9;
- clinical applicability (noted in the following with CA) - questions 8 and 14;
- relevancy (noted in the following with R) - questions 11 and 13;

Some of the questions regarding specific aspects were inverted in the questionnaire to make sure that the participants didn't mistakenly check a different answer (e.g. questions 11 and 13).

For the evaluation of the answers the five-point Likert scale was substituted with numerical values from 1 to 5 (NO = 1, no = 2, uncertain = 3, yes = 4 and YES = 5). The boxplots in Figure 6.7 are providing an overview regarding the feedback

obtained from the participants regarding all the evaluated usability metrics and the mean values for each answer from left to right.

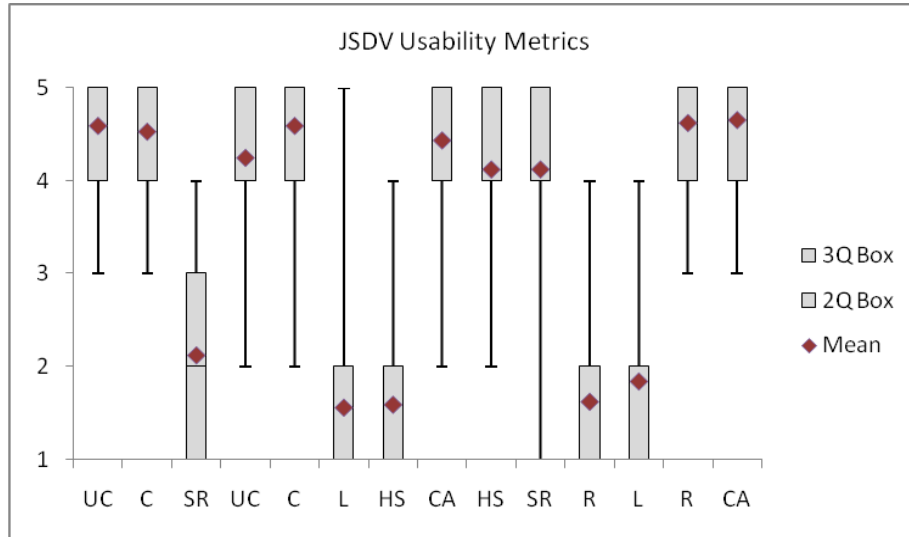


Figure 6.7. JSDV Usability metrics

The boxplots in Figure 6.7 are showing that the participants had strong opinions regarding the various usability aspects. The answers regarding SR show that there was a difference based on how the question was asked. "I think the software responds too slowly at certain stages" (question 3) was regarded differently by the participants from "The response time of the software is really fast" (question 10). The respondents were not sure if the software responded too slowly, but they were sure it was really fast.

The participants satisfaction when using the 2D mouse was evaluated using a 13 questions questionnaire (annex 6) based on the ISO 9241-9 assessment of comfort questionnaire.

The questions used, scale used for evaluation, averages and standard deviation regarding the answers of the entire population (N = 32) are shown in Table 6.5.

Table 6.5 JSDV 2D mouse assesment questions, scale, averages and standard deviation.

Question	Scale	Average	SD
1. The operation speed of the mouse was	1 - 5 too fast - too slow	2.28	0.84
2. Finger fatigue due to operating the device was	1 - 5 none - very high	2.13	1.45
3. Accurate pointing on the screen was	1 - 5 easy - difficult	1.53	0.71

4. The force required for actuation was	1 - 5 too low - too high	2.44	0.97
5. The mental effort required for operation was	1 - 5 too low - too high	2.41	0.82
6. The physical effort required for operation was	1 - 5 too low - too high	2.13	1.02
7. Smoothness during operation was	1 - 5 very smooth - very rough	1.88	0.86
8. Wrist fatigue during operation was	1 - 5 none - very high	1.63	0.96
9. Arm fatigue due to the operation was	1 - 5 very high - none	4.50	0.79
10. Shoulder fatigue due to operation was	1 - 5 none - very high	1.44	0.83
11. Neck fatigue due to operation was	1 - 5 very high - none	4.69	0.63
12. General comfort	1 - 5 very uncomfortable - very comfortable	4.06	1.12
13. Overall, the 2D mouse was	1 - 5 very easy to use - very difficult to use	1.38	0.70

The boxplots in Figure 6.8 are providing an overview of the answers for each question and the mean values obtained for the entire population (N = 32).

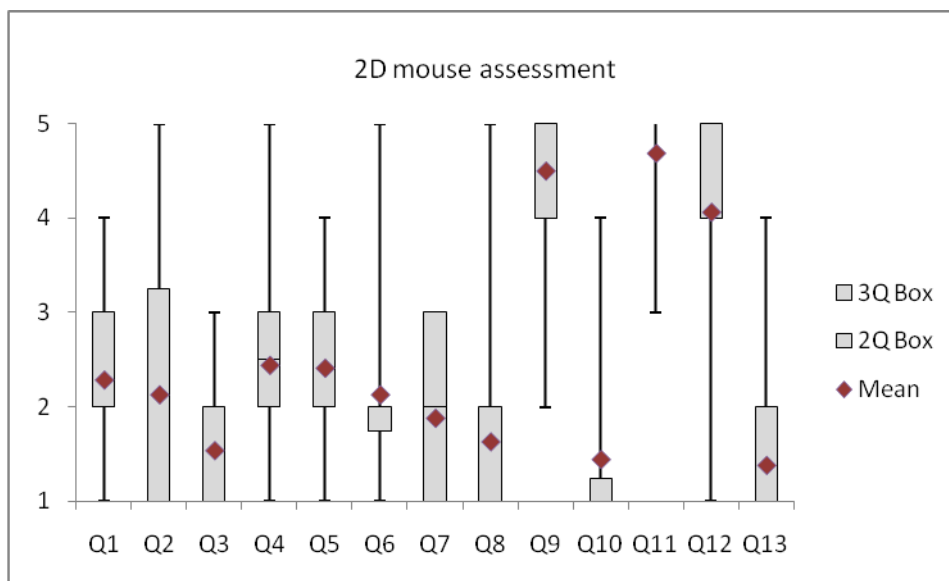


Figure 6.8. JSDV 2D mouse assessment

The average results for questions 4 and 5 show that the users were not very determined regarding the *force required for actuation* and the *mental effort required for operation*.

The subjective metrics regarding the ease of use and satisfaction regarding the LEAP Motion controller was assessed using a 13 questions form (annex 7) also based on the ISO 9241-9 assessment of comfort questionnaire.

The Table 6.6 provides the details regarding the questions, scale used for evaluation, averages and standard deviation based on the answers of the entire population (N = 32).

Table 6.6 JSDV LEAP Motion assesment questions, scale, averages and standard deviation.

Question	Scale	Average	SD
1. The operation speed of the LEAP Motion Controller was	1 - 5 too fast - too slow	2.66	0.96
2. Finger fatigue due to operating the LEAP device was	1 - 5 none - very high	1.88	1.19
3. Accurate pointing on the screen was	1 - 5 easy - difficult	2.19	0.98
4. The force required for actuation was	1 - 5 too low - too high	2.41	1.03
5. The mental effort required for operation was	1 - 5 too low - too high	2.31	0.85
6. The physical effort required for operation was	1 - 5 too low - too high	1.91	0.63
7. Smoothness during operation was	1 - 5 very smooth - very rough	1.97	0.92
8. Wrist fatigue during operation was	1 - 5 none - very high	1.72	0.94
9. Arm fatigue due to the operation was	1 - 5 very high - none	4.19	0.92
10. Shoulder fatigue due to operation was	1 - 5 none - very high	1.75	1.12
11. Neck fatigue due to operation was	1 - 5 very high - none	4.56	0.90
12. General comfort	1 - 5 very uncomfortable - very comfortable	4.19	0.88
13. Overall, the LEAP Motion Controller was	1 - 5 very easy to use - very difficult to use	1.97	0.85

The boxplots in Figure 6.9 are showing the distribution of the answers for each question and the mean values obtained for the entire population (N = 32).

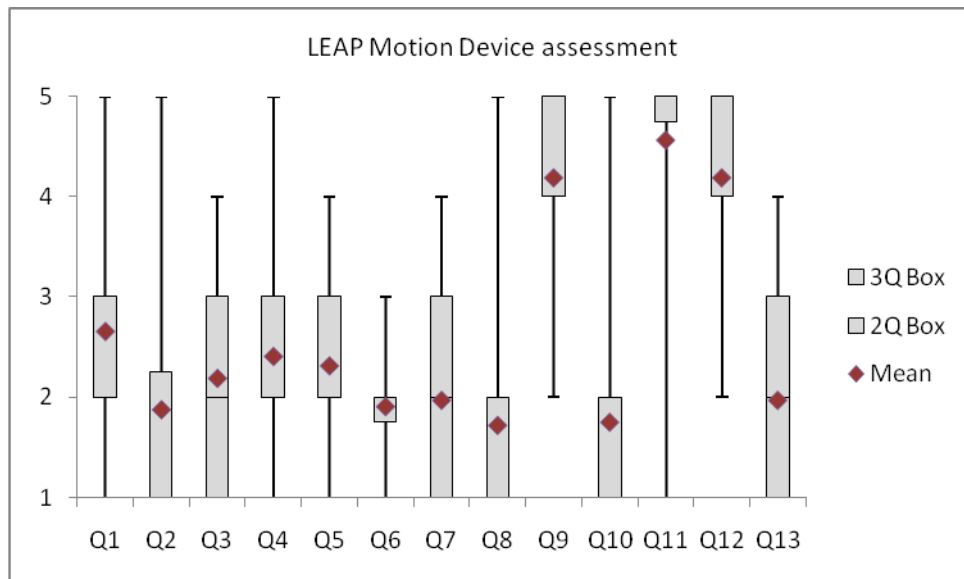


Figure 6.9. JSDV LEAP Motion assessment

The average results for question 1 shows that the participants had mixed opinions regarding the *operation speed* of the LEAP Motion controller due to the fact that only 6.25 % of them had a previous experience with this gesture tracking device. For the rest of the 93.75 % of the participants, besides the fact that this was their first encounter with the JSDV application, they also had no previous experience with a gesture tracking device. The answers collected for questions 4 and 5 again showed that the users were not very determined regarding the *force required for actuation* and the *mental effort required for operation* same as in the case of the 2D mouse.

Since the questions regarding the 2D mouse and the LEAP Motion controller were similar the answers provided regarding both devices are compared in Table 6.7.

Table 6.7 Comparison of the device assessment results for 2D mouse and LEAP Motion for JSDV

Question	Scale	Mean 2D mouse	SD 2D mouse	Mean LEAP Motion	SD LEAP Motion
1. The operation speed of the mouse/LEAP Motion Controller was	1 - 5 too fast - too slow	2.28	0.84	2.66	0.96
2. Finger fatigue due to operating the device/LEAP device was	1 - 5 none - very high	2.13	1.45	1.88	1.19

110 6. Evaluation

3. Accurate pointing on the screen was	1 - 5 easy - difficult	1.53	0.71	2.19	0.98
4. The force required for actuation was	1 - 5 too low - too high	2.44	0.97	2.41	1.03
5. The mental effort required for operation was	1 - 5 too low - too high	2.41	0.82	2.31	0.85
6. The physical effort required for operation was	1 - 5 too low - too high	2.13	1.02	1.91	0.63
7. Smoothness during operation was	1 - 5 very smooth - very rough	1.88	0.86	1.97	0.92
8. Wrist fatigue during operation was	1 - 5 none - very high	1.63	0.96	1.72	0.94
9. Arm fatigue due to the operation was	1 - 5 very high - none	4.50	0.79	4.19	0.92
10. Shoulder fatigue due to operation was	1 - 5 none - very high	1.44	0.83	1.75	1.12
11. Neck fatigue due to operation was	1 - 5 very high - none	4.69	0.63	4.56	0.90
12. General comfort	1 - 5 very uncomfortable - very comfortable	4.06	1.12	4.19	0.88
13. Overall, the 2D mouse/LEAP Motion Controller was	1 - 5 very easy to use - very difficult to use	1.38	0.70	1.97	0.85

When interpreting the results from Table 6.7 the scale used for each answer needs to be considered, since the values were mixed to make sure the participants read the questions properly.

The chart in Figure 6.10 shows side by side the averages of the answers regarding the 2D mouse and the LEAP Motion Device assessment. The answers were mixed, in some cases the participants thought that the mouse performed better than the gesture tracking device and in other cases quite the opposite. Overall the results were similar in both cases showing that the users could interact with the application even if they were not accustomed with the LEAP Motion Device.

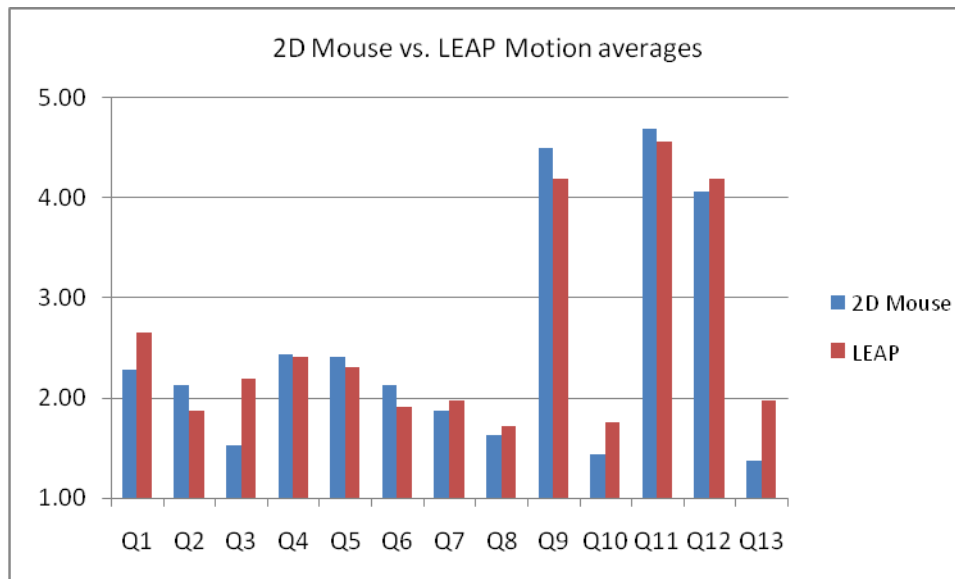


Figure 6.10. JSDV 2D mouse vs. LEAP Motion assessment

Some more prominent differences might be observed regarding the answers for question 1, 3, 10 and 13. The first question dealt with the *operation speed* of the device and the users considered the mouse faster. In case of the third question the participants on average considered that the *accuracy of pointing on the screen* was more difficult to achieve with the LEAP Motion. As expected in case of question 10 *shoulder fatigue experienced due to operation* was slightly higher in the case of the gesture tracking device. Even if on average the participants considered the LEAP Motion more *difficult to use* (as shown by their average opinions expressed regarding question 13) they considered the gesture tracking device more *comfortable* (based on the average answers received for question 12).

The participants considered that the *mental effort required for operation* was slightly higher on average in the case of the 2D mouse (based on the answers received for question 5). This supports the statement that they felt more naturally when they used a gesture-based interaction as it should be the case of any natural user interface. This is especially important since 93.75 % of the participants never used a LEAP Motion controller before and they had to learn for the first time what gestures they need to use in order to separate a specific slice from the rendered 3D stack and to move it away. Since the gestures used resemble the real gestures (like pointing towards something and using two hands to set an object aside) the participants felt that the gestures were already familiar to them.

Overall the JSDV application was well received by the participants and they found it very engaging. 96 % of the medical students (23 out of 24) expressed their positive opinion that they would like to use the software at their workplace and that they would recommend it to their colleagues.

#### **6.4. Usability evaluation overview for the GLmol LEAP application**

The purpose of the usability evaluation for the GLmol LEAP application was to appraise user performance and user-satisfaction levels that would help identify potential problems within the applications' user interface after it has been extended for gesture-based control.

The usability testing took place in a controlled testing environment with the help of a representative group of volunteers and included the testing of the following functionalities of the GLmol LEAP application:

- locating the help menu;
- loading of a PDB file of STRUCTURE OF PORIN REFINED AT 1.8 ANGSTROMS RESOLUTION (publicly available sample is obtained from <http://www.rcsb.org/pdb/explore/explore.do?structureId=2POR>);
- mouse-based interaction with the rendered 3D model of the chemical structure (rotation);
- gesture-based interaction with the 3D model by using a LEAP Motion Controller (zoom);

The tasks that had to be executed by the participants were established by the author of the thesis for covering the most representative functionalities provided by the GLmol LEAP application, due to the short timeframe for which the participants were usually available (approximately 10 minutes). The tasks needed to perform were the same for all the participants and the task scenarios were the following:

- Find out how to start using the GLmol LEAP application;
- Locate and open a local PDB file for visualization;
- Rotate the rendered 3D model with the mouse;
- Enlarge the 3D model with the help of the LEAP Motion device;

The task scenarios, successful completion criteria, procedures required to perform and a maximum reference time for completion were described in annex 3 of the test plan. A separate time on task table (annex 4) was used to record user performance such as the time for completion per task for each user, as well as eventual critical and non-critical errors for later processing. This provided answers regarding successful completion rates, error-free rates and time on task. The completion rates were calculated as the percentage of users that completed the task without critical errors. Error-free rates were calculated as the percentage of participants who had no errors at all (neither critical nor non-critical errors).

Three separate questionnaires (annex 5, 6 and 7) have been used to evaluate the usability metrics that provide insights regarding the ease of use of the application and user satisfaction when confronted with a LEAP Motion against a 2D mouse.

After the completion of the task scenarios described above, a five-point Likert scale usability questionnaire consisting of 14 questions (annex 5) was used to record answers from each participant regarding the following quantitative metrics:

- consistency;
- learnability;
- relevancy;



- domain applicability;
- user control and freedom;
- software response speed;
- help section related issues;

The subjective metrics were evaluated with the help of two questionnaires (annex 6 and annex 7) based on the ISO 9241-9 assessment of comfort questionnaire. These were meant to provide answers regarding the participants satisfaction when using a 2D mouse or a LEAP Motion Controller to manipulate the 3D model of the chemical structure.

### 6.5. Methodology for testing the GLmol LEAP application

The user group that participated in the usability test consisted of 11 university professors (specialized in organic chemistry) and 12 chemistry students all from the Industrial Chemistry and Environmental Engineering faculty of the Politehnica University of Timisoara. The testing took place at the Industrial Chemistry and Environmental Engineering faculty of the Politehnica University of Timisoara.

For the testing the author of the thesis used a Lenovo ideapad Y510P laptop, that had an Intel Core i7-4700MQ CPU at 2.40 GHz, 8GB RAM, an Nvidia GeForce GT755M graphics card and a display resolution of 1920x1080 at 60 Hz, together with a custom built autostereoscopic 3D display in conjunction with an Asus monitor at a resolution of 1920 x 1080 pixels and a LEAP Motion Controller. The laptop had a Windows 10 Education 64-bit operating system, Google Chrome v 56.0.2924.87, Mozilla Firefox v 51.0.1 browsers and LEAP Motion software ver. 2.3.1. Figure 6.11 shows the entire setup used to test the 3D interface of the GLmol LEAP application.



Figure 6.11. Setup used for testing the 3D user interface of the GLmol LEAP application

Before the test each participant completed a consent form regarding their willingness to participate (annex 1) and a demographic and background information questionnaire (annex 2).

After the test the participants also completed a usability questionnaire (annex 5) and two separate device assessment of comfort questionnaires for the 2D mouse version (annex 6) and for the LEAP Motion (annex 7).

A detailed description regarding the participants, training and specific procedures will be provided in the following.

### 6.5.1. Participants

The GLmol LEAP application has been tested by 23 volunteers (7 male, 16 female). The participants' ages ranged between 20 and 57 year old, with an average of 31.22 years old, distributed as in Figure 6.12.

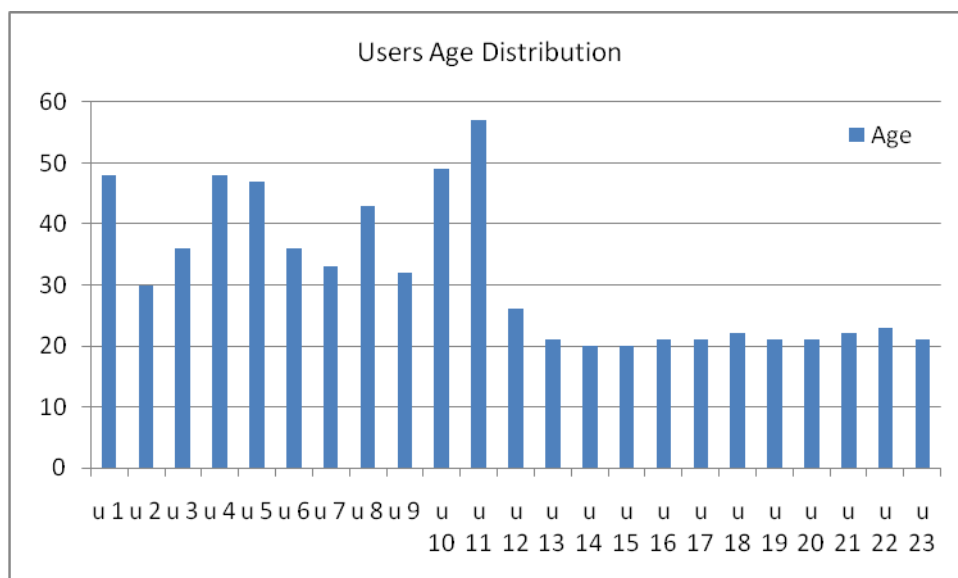


Figure 6.12. GLmol LEAP users age distribution

Their educational background ranged from 12 chemistry students to 11 university professors. Three participants had a specialization in computational chemistry and one in macromolecular chemistry.

All participants declared that they had previous experience using a computer (between 8 and 30 years). Only thirteen participants declared that they had a previous experience with 3D molecular viewers (from one to twenty years). Seven participants declared that they had a previous experience with 3D molecular datasets (between 3 and 20 years). 21 out of 23 participants declared that they are comfortable using a web browser. While 22 of the participants declared that they are comfortable using a 2D mouse, only one of them had a previous experience with a LEAP Motion Controller. 22 out of 23 participants were right handed and one left

handed. Only five participants (all from the students group) declared that they had a previous experience with a 3D pseudo-holographic display.

### **6.5.2. Training**

The participants had a brief overview of the usability test procedure and had a ten minutes training regarding the application and apparatus used. The volunteers were divided into smaller groups and each group had a separate training session to ensure that every one of them can see the user interface details and understands the role of each device used for testing the application.

After answering the eventual questions regarding the equipment and the application the users were asked to complete the task scenarios without any kind of external aid from the test moderator.

### **6.5.3. Procedure**

In order to have a controlled environment and to obtain consistent results all the usability testing sessions took place at the Industrial Chemistry and Environmental Engineering faculty of the Politehnica University of Timisoara. The participants were divided into smaller groups due to the estimated time of 10 minutes for each volunteer needed to complete the questionnaires and the test scenarios. The participant's interaction with the GLmol LEAP application was monitored by the author of the thesis (who assumed the role of the test moderator) seated next to them.

The author of the thesis instructed the participants regarding the procedures, apparatus used and the GLmol LEAP application, emphasizing that they are evaluating the application, rather than the moderator evaluating them in order to alleviate any stress factors that might have influenced the test results. After that each participant signed a consent form (annex 1) confirming that they participation is voluntary and that they might raise any concerns during the test session. The volunteers also completed a pre-test demographic and background information questionnaire (annex 2) regarding their age, gender, professional status, specialty, previous experience with computers, 3D molecular viewers, 3D molecular datasets, web browser usage, handedness and the equipment used for testing (2D mouse, LEAP Motion Device, Pseudo-holographic 3D display). The author of the thesis explained the participants that he will measure the amount of time taken to complete each task scenario and that they should not have any other exploratory behavior outside the ones required to complete the task excepting the fact that they were allowed and encouraged to use the application's help section whenever they felt the need to do so. After that the moderator read out loud each task (from annex 3) and instructed the volunteers to start. The tasks were considered finished either if the participant indicated so, either if they exceeded the previously estimated time for completion.

The times for completion, eventual critical and non-critical errors were registered in a separate form (annex 4) for post processing. The non-critical errors were recorded in cases when the participant completed by himself the task without any kind of guidance from the moderator, but did not find the needed functionalities right away. The critical errors might have been the result of not being able to complete the task or exceeding the estimated time for completion by the

participants. Although there were no critical errors these were planned to be recorded as the reference time plus one second (e.g. for 60 to be recorded as 61) if they happened.

After the completion of the tasks the participants were asked to complete the post-test usability questionnaire (annex 5) and the device assessment of comfort questionnaire for the 2D mouse (annex 6) and for the LEAP Motion Controller (annex 7).

### 6.6. Test results for the GLmol LEAP application

The results of the evaluation of the GLmol LEAP application revealed that on average the participants completed all the tasks under the previously estimated reference times. Table 6.8 presents the reference times for each task, mean (average), the minimum, maximum times for completion and the population ( $N = 23$ ) standard deviation (SD).

Table 6.8 GLmol LEAP time on task results showing the reference, average, minimum, maximum times and standard deviation

<b>Task no.</b>	<b>T 1</b>	<b>T 2</b>	<b>T 3</b>	<b>T 4</b>
<b>Reference time (s)</b>	60	60	30	30
<b>Average (s)</b>	11.87	10.96	1.70	6.74
<b>Min time (s)</b>	3	6	1	1
<b>Max time (s)</b>	43	30	3	24
<b>Population SD (s)</b>	8.48	5.70	0.62	5.80

The average times for completion were close to the population average in both cases of the university professors ( $n = 11$ ) and the chemistry students ( $n = 12$ ). The t-test results showed no conclusive differences between the times recorded for tasks 1 and 2. Table 6.9 shows the average times for completion, unbiased (corrected) sample variances and sample standard deviations for both user groups as well as the p-values obtained for a two-tailed test where the two samples had unequal variance. Bessel's correction had been used to calculate the sample variances ( $n-1$  was used instead of  $n$  in the calculations). The university professors are noted with CP and the chemistry students with CS in the table.

Table 6.9 GLmol LEAP average times on task, corrected sample variances, sample standard deviations and t-test results for the two user groups

<b>Task no.</b>	<b>T 1</b>	<b>T 2</b>	<b>T 3</b>	<b>T 4</b>
-----------------	------------	------------	------------	------------

<b>Reference time (s)</b>	60	60	30	30
<b>Average CP (s)</b>	10.73	10.82	1.36	3.36
<b>Corrected sample variance CP</b>	46.22	43.76	0.45	13.45
<b>Sample SD CP (s)</b>	11.31	14.14	0.71	0.71
<b>Average CS (s)</b>	12.92	11.08	2.00	9.83
<b>Corrected sample variance CS</b>	105.72	28.08	0.18	36.15
<b>Sample SD CS (s)</b>	9.90	4.24	0.71	2.83
<b><i>p</i>-values</b>	0.551	0.917	0.016	0.006

The differences between the averages of the two user groups recorded for task 4 were produced by the fact that most of the participants from the chemistry students group forgot to raise their hands to an appropriate height from the LEAP Motion Controller in order to be detected (as it was mentioned during the training phase). Immediately after raising their hands they were able to interact with the displayed 3D model on their own. The author of the thesis didn't consider this as an exploratory behavior since most of them had no previous experience with such a device.

Table 6.10 shows the recorded times per task scenario for the entire population (N = 23). The only one recorded non-critical error is marked with bolded text.

Table 6.10 Recorded times per task scenarios for GLmol LEAP

<b>Task no.</b>	<b>T 1</b>	<b>T 2</b>	<b>T 3</b>	<b>T 4</b>
<b>Reference time (s)</b>	60	60	30	30
<b>User 1</b>	22	30	2	3
<b>User 2</b>	17	10	1	2
<b>User 3</b>	7	6	1	2
<b>User 4</b>	7	12	1	2

<b>User 5</b>	11	7	1	14
<b>User 6</b>	4	<b>11</b>	2	2
<b>User 7</b>	19	8	1	1
<b>User 8</b>	5	9	1	2
<b>User 9</b>	3	7	1	5
<b>User 10</b>	17	9	3	2
<b>User 11</b>	6	10	1	2
<b>User 12</b>	20	16	1	2
<b>User 13</b>	7	20	2	24
<b>User 14</b>	10	10	2	15
<b>User 15</b>	10	11	2	10
<b>User 16</b>	7	8	2	7
<b>User 17</b>	9	21	2	11
<b>User 18</b>	16	6	2	9
<b>User 19</b>	8	12	2	16
<b>User 20</b>	10	7	2	7
<b>User 21</b>	9	6	2	6
<b>User 22</b>	43	6	3	5
<b>User 23</b>	6	10	2	6

The separate completion and error-free rates were calculated for each task in order to determine the user performance based on the time for completion for each individual task.

Same as in the case of the JSDV the completion rate shows the percentage of participants that completed the task with no critical errors, while error-free rates are showing the percentage of users who had no errors at all during the completion of the task. Table 6.11 shows the separate completion rates (CR) and error-free rates (EFR) for the entire population, for the university professors (noted with CP) and for the chemistry students (noted with CS). The last column shows the average completion rates and error-free rates for all the tasks.

Table 6.11 Completion rates and error-free rates for GLmol LEAP

Task no.	T 1	T 2	T 3	T 4	Average
<b>CR %</b>	100.00	100.00	100.00	100.00	<b>100.00</b>
<b>EFR %</b>	100.00	95.65	100.00	100.00	<b>98.91</b>
<b>CP CR %</b>	100.00	100.00	100.00	100.00	<b>100.00</b>
<b>CP EFR %</b>	100.00	90.91	100.00	100.00	97.73
<b>CS CR %</b>	100.00	100.00	100.00	100.00	100.00
<b>CS EFR %</b>	100.00	100.00	100.00	100.00	100.00

The recorded high averages for the entire population (N = 23) of all the tasks regarding completion rates (CR = 100.00 %) and error-free rates (EFR = 98.91 %) are leading to the conclusion that the users performance was exceptional since this was their first time when they used the GLmol LEAP application. These averages are very high for the group of the university professors (n = 11) and even higher in the case of the chemistry students (n = 12) where both the average completion rate and the average error-free were 100.00 %.

The chart in Figure 6.13 shows the average completion rates for the entire population, for the university professors group (n = 11) and for the chemistry students group (n = 12).

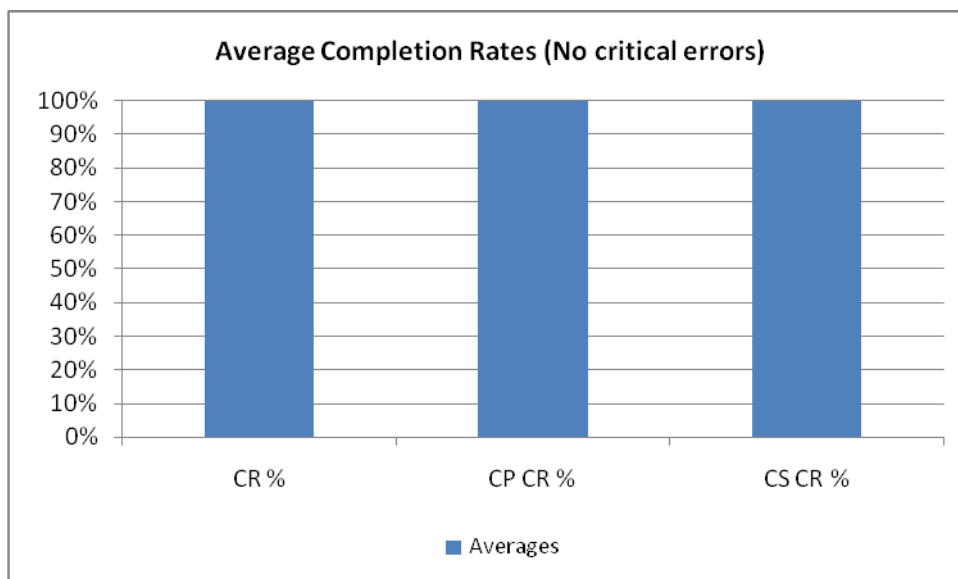


Figure 6.13. GLmol LEAP average completion rates

As the chart shows there were no critical errors recorded during the evaluation since all the participants completed all the tasks in the estimated time.

The chart in Figure 6.14 provides an insight regarding the different error-free rates regarding the entire population (N = 23), for the university professors (n = 11) and for the chemistry students (n = 12).

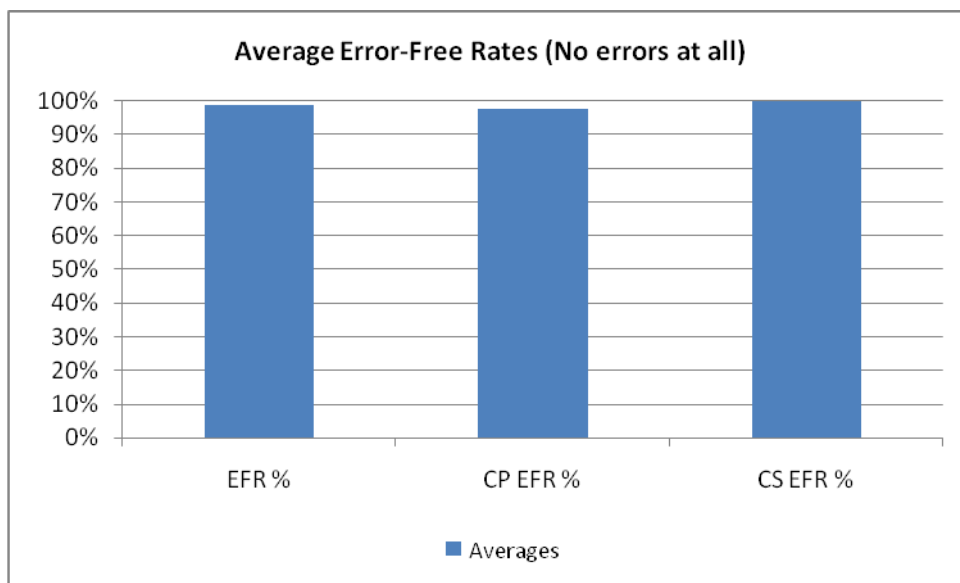


Figure 6.14. GLmol LEAP average error-free rates

The chart in Figure 6.14 covers all the errors (critical and non-critical) for all the evaluated task scenarios.

The high average completion rates (100.00 %) and error-free rates (98.91 %) for all the tasks considering the entire population (N = 23) revealed the overall success from the point of view of the user performance of the GLmol LEAP application.

The usability metrics were assessed using the annex 5 where the users provided feedback regarding their views upon the following affirmations:

1. I feel comfortable using this software.
2. The information presented on the screen is clear enough to understand how to operate the GLmol LEAP application.
3. I think the software responds too slowly at certain stages.
4. I have enough information to operate the application.
5. I think the user interface is organized in a simple and consistent manner.
6. Learning to operate the software was too hard.
7. I think that the help section is not very useful.
8. I would like to use this software at my workplace.
9. It was easy to find out what to do next when I got stuck.
10. The response time of the software is really fast.
11. There is no relevant information on the screen in order to operate the software.
12. There is too much information that I have to understand before I can operate the application.



13. The way the buttons are organized is logical and relevant.

14. I would recommend this software to my colleagues.

similar as in case of the JSDV application and adapted to the current situation.

For the possible responses a five-point Likert scale (ranging between YES, yes, uncertain, no and NO) has been used. The next usability aspects were evaluated:

- user control and freedom (noted in the following with UC) - questions 1 and 4;
- consistency (noted in the following with C) - questions 2 and 5;
- software response speed (noted in the following with SR) - questions 3 and 10;
- learnability (noted in the following with L) - questions 6 and 12;
- help section related issues (noted in the following with HS) - questions 7 and 9;
- domain applicability (noted in the following with DA) - questions 8 and 14;
- relevancy (noted in the following with R) - questions 11 and 13;

Same as in the case of the JSDV application, some of the questions regarding specific aspects were inverted in the questionnaire to make sure that the participants didn't mistakenly check a different answer (e.g. questions 11 and 13).

For the evaluation of the answers the five-point Likert scale was substituted with numerical values from 1 to 5 (NO = 1, no = 2, uncertain = 3, yes = 4 and YES = 5). The boxplots in Figure 6.15 are providing an overview regarding the feedback obtained from the participants regarding all the evaluated usability metrics and the mean values for each answer from left to right.

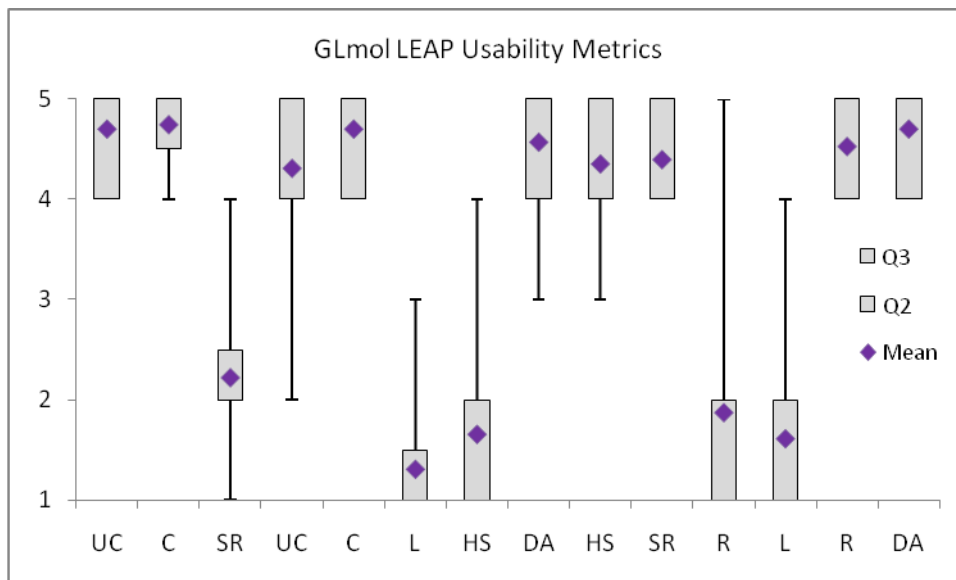


Figure 6.15. GLmol LEAP Usability metrics

Since the mean values for the boxplots in Figure 6.15 are mostly distributed in the 1 to 2 and 4 to 5 ranges the author of the thesis concluded that the participants had strong opinions regarding the various usability aspects, excepting the case for the first question regarding the SR. Similar as for the JSDV evaluation, the answers regarding SR were based on how the question was asked. "I think the software responds too slowly at certain stages" (question 3) was perceived differently by the participants from "The response time of the software is really fast" (question 10). The participants were not sure if the software responded too slowly (question 3), but they were convinced that it was really fast.

The participant's satisfaction regarding the use of the 2D mouse was evaluated using a 13 questions questionnaire (annex 6) based on the ISO 9241-9 assessment of comfort questionnaire.

The questions used, scale used for evaluation, averages and standard deviation regarding the answers of the entire population (N = 23) are shown in Table 6.12.

Table 6.12 GLmol LEAP 2D mouse assesment questions, scale, averages and standard deviation.

Question	Scale	Average	SD
1. The operation speed of the mouse was	1 - 5 too fast - too slow	2.30	0.62
2. Finger fatigue due to operating the device was	1 - 5 none - very high	2.00	1.14
3. Accurate pointing on the screen was	1 - 5 easy - difficult	1.74	0.79
4. The force required for actuation was	1 - 5 too low - too high	2.78	0.78
5. The mental effort required for operation was	1 - 5 too low - too high	2.13	0.61
6. The physical effort required for operation was	1 - 5 too low - too high	1.91	0.58
7. Smoothness during operation was	1 - 5 very smooth - very rough	2.48	0.97
8. Wrist fatigue during operation was	1 - 5 none - very high	1.61	0.77
9. Arm fatigue due to the operation was	1 - 5 very high - none	4.09	1.18
10. Shoulder fatigue due to operation was	1 - 5 none - very high	1.48	0.58
11. Neck fatigue due to operation was	1 - 5 very high - none	4.04	1.40
12. General comfort	1 - 5 very uncomfortable - very comfortable	4.57	0.88
13. Overall, the 2D mouse was	1 - 5 very easy to use - very difficult to use	1.22	0.41

The boxplots in Figure 6.16 are providing an overview of the answers for each question and the mean values obtained for the entire population (N = 23).

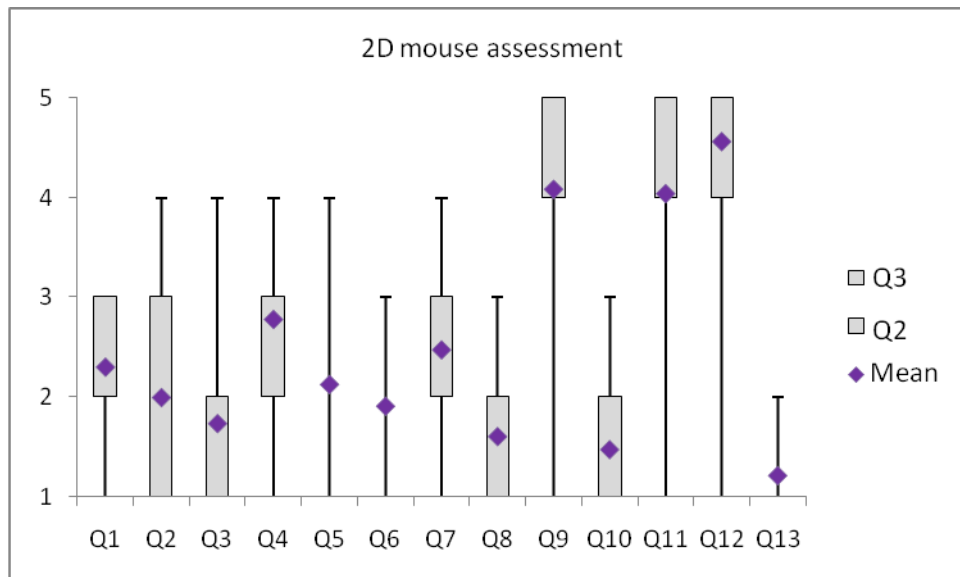


Figure 6.16. GLmol LEAP 2D mouse assessment

The average results for questions 4 and 7 show that the users were not very determined regarding the *force required for actuation* and the *smoothness during operation*.

The subjective metrics regarding the ease of use and satisfaction regarding the LEAP Motion controller was assessed using a 13 questions form (annex 7) also based on the ISO 9241-9 assessment of comfort questionnaire.

The Table 6.13 provides the details regarding the questions, scale used for evaluation, averages and standard deviation based on the answers of the entire population (N = 23).

Table 6.13 GLmol LEAP application LEAP Motion assesment questions, scale, averages and standard deviation.

Question	Scale	Average	SD
1. The operation speed of the LEAP Motion Controller was	1 - 5 too fast - too slow	2.48	0.65
2. Finger fatigue due to operating the LEAP device was	1 - 5 none - very high	2.00	1.02
3. Accurate interaction with the 3D model rendered on the screen was	1 - 5 easy - difficult	1.96	0.95
4. The force required for actuation was	1 - 5 too low - too high	2.52	0.93
5. The mental effort required for operation was	1 - 5 too low - too high	2.00	0.72
6. The physical effort required for operation was	1 - 5 too low - too high	1.83	0.70

124 6. Evaluation

7. Smoothness during operation was	1 - 5 very smooth - very rough	2.39	0.97
8. Wrist fatigue during operation was	1 - 5 none - very high	1.78	0.83
9. Arm fatigue due to the operation was	1 - 5 very high - none	4.22	0.72
10. Shoulder fatigue due to operation was	1 - 5 none - very high	1.83	1.05
11. Neck fatigue due to operation was	1 - 5 very high - none	4.52	0.65
12. General comfort	1 - 5 very uncomfortable - very comfortable	4.65	0.56
13. Overall, the LEAP Motion Controller was	1 - 5 very easy to use - very difficult to use	1.52	0.58

The boxplots in Figure 6.17 are showing the distribution of the answers for each question and the mean values obtained for the entire population (N = 23).

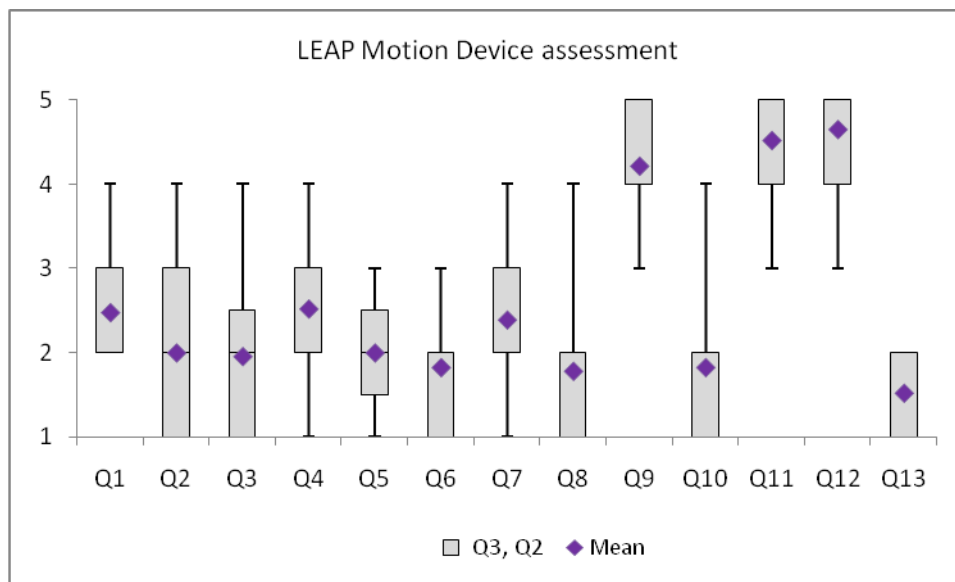


Figure 6.17. GLmol LEAP application LEAP Motion assessment

The average results for question 1 showed that the participants were not very determined regarding the *operation speed* of the LEAP Motion controller however they leaned towards a positive answer. The answers regarding the questions 4 and 7, same as in the case of the 2D mouse, revealed that the users were not very determined regarding the *force required for actuation* and the *smoothness during operation* with the tendency being towards a small force needed for actuation and that the controller was operated smoothly.

As in the case of the JSDV evaluation, since the questions regarding the 2D mouse and the LEAP Motion controller were similar the answers provided regarding both devices were compared and provided in Table 6.14.

Table 6.14 Comparison of the device assessment results for 2D mouse and LEAP Motion for GLmol LEAP

Question	Scale	Mean 2D mouse	SD 2D mouse	Mean LEAP Motion	SD LEAP Motion
1. The operation speed of the mouse/LEAP Motion Controller was	1 - 5 too fast - too slow	2.30	0.62	2.48	0.65
2. Finger fatigue due to operating the device/LEAP device was	1 - 5 none - very high	2.00	1.14	2.00	1.02
3. Accurate pointing/ interaction with the 3D model rendered on the screen was	1 - 5 easy - difficult	1.74	0.79	1.96	0.95
4. The force required for actuation was	1 - 5 too low - too high	2.78	0.78	2.52	0.93
5. The mental effort required for operation was	1 - 5 too low - too high	2.13	0.61	2.00	0.72
6. The physical effort required for operation was	1 - 5 too low - too high	1.91	0.58	1.83	0.70
7. Smoothness during operation was	1 - 5 very smooth - very rough	2.48	0.97	2.39	0.97
8. Wrist fatigue during operation was	1 - 5 none - very high	1.61	0.77	1.78	0.83
9. Arm fatigue due to the operation was	1 - 5 very high - none	4.09	1.18	4.22	0.72
10. Shoulder fatigue due to operation was	1 - 5 none - very high	1.48	0.58	1.83	1.05
11. Neck fatigue due to operation was	1 - 5 very high - none	4.04	1.40	4.52	0.65
12. General comfort	1 - 5 very uncomfortable - very comfortable	4.57	0.88	4.65	0.56
13. Overall, the 2D mouse/LEAP Motion Controller was	1 - 5 very easy to use - very difficult to use	1.22	0.41	1.52	0.58

As in the case of the JSDV evaluation, when interpreting the results shown in Table 6.14 the scale used for each answer needs to be considered, since the possible answers were mixed to observe if the participants properly read all the questions.

The chart in Figure 6.18 provides a comparison between the averages of the answers provided regarding the 2D mouse and the LEAP Motion Device assessment. The results are showing that on some occasions the participants found the 2D mouse easier to use (since they were using one on a daily basis), but they also considered the LEAP Motion Controller more comfortable. This is a very important

aspect that needs to be considered since the vast majority of them never used this gesture tracking device before, but were still able to complete the associated task.

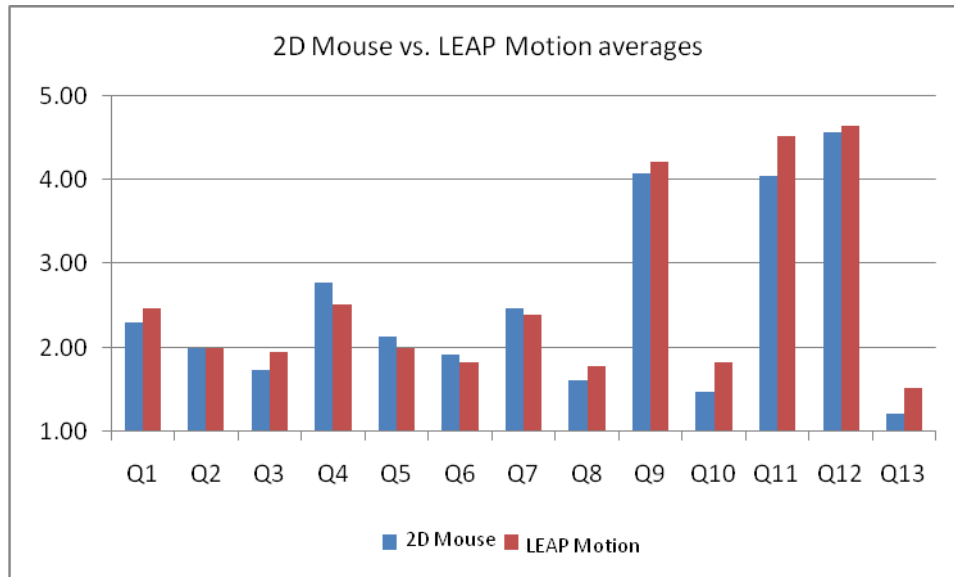


Figure 6.18. GLmol LEAP 2D mouse vs. LEAP Motion assessment

Even if the graphics in the Figure 6.18 are showing similar tendencies regarding the answers in general, there are some observable differences between the answers provided for question 4, 10, 11 and 13. Question 4 was regarding the *force required for actuation* and the participants considered that it was higher in the case of the 2D mouse as it was expected. *Shoulder fatigue* (question 10) was considered higher in the case of the LEAP Motion device leading towards a similar tendency as in the case of the JSDV evaluation results. Based on the answers received for the question 11 regarding *neck fatigue*, it seems that this was higher in the case of the 2D mouse. As it has been expected the participants considered the LEAP Motion slightly *more difficult to use* considering the average answers for question 13, but they also considered it *smoother during operation* based on the answers provided for question 7.

Based on the average answers received for question 5, the participants considered that the *mental effort required for operation* was slightly higher in the case of the 2D mouse, showing a similar tendency as in the results of the JSDV application's evaluation. This tendency strengthens the fact that the users appreciated the gesture-based interaction to be more natural even if 95.65 % of the participants had no previous experience with a LEAP Motion controller.

Based on the results, all the participants (100 %) appreciated that they would recommend the GLmol LEAP application to their colleagues and 91.30 % of the participants also considered that they would like to use the application at their workplace.

## **6.7. Conclusions**

The testing followed the same patterns for both JSDV and GLmol LEAP applications. The author of the thesis divided in both cases the entire population into smaller sample groups for comparison.

In both cases the results showed that the natural user interface was well received and helped the participants to obtain better results even when they were using a novel gesture-tracking device like the LEAP Motion.

The results for testing the GLmol LEAP application were also encouraging and provided a viable option for future virtual laboratories when multimodal interaction can act as a key component to disseminate chemistry related knowledge.

## 7. CONCLUSIONS AND FURTHER DEVELOPMENT

The research subject of this thesis is strongly related to the latest evolutions of the novel input and output devices that need a new approach regarding the development of the next generation user interfaces. As the author demonstrated through the current thesis the novel gesture tracking technologies and 3D autostereoscopic devices can provide a significant improvement resulting in viable natural user interfaces.

The theoretical developments during the research are reflected in the development of two software applications providing a basis for more complex educational virtual laboratories targeting the medical and chemistry students.

The thesis ensures a strong base for further research by providing answers regarding the novel hardware and software technologies discussed that had been obtained reaching the thesis objectives:

- a. A comprehensive analysis of the present hardware technologies used in the development of state-of-the-art natural user interfaces. These were grouped into gesture-tracking devices, brain-computer interfaces, several display technologies that can be used in 3D pseudo-holographic imaging and haptic devices used for tactile feedback.
- b. An extensive analysis of the novel software technologies focusing on open source solutions that can provide viable alternatives for gesture tracking and 3D rendering. The presented solutions eliminate the limitations imposed by some of the proprietary software components that could limit the activities of a research group.
- c. Modelling and development of a browser-based DICOM viewer that allows the users to interact with the help of simple and natural gestures with the rendered 3D model that has been visualized on a 3D pseudo-holographic display.
- d. Development of a segmentation algorithm that for a browser-based environment overriding the speed limitations exposed in chapter 4.
- e. Implementation of an interactive client-side colorization algorithm to isolate the relevant anatomical details in the rendered 2D medical images for the multiplanar 3D reconstruction.
- f. Extension of a browser-based 3D molecular viewer by adding gesture tracking functionalities providing enhanced multimodal user interaction capabilities.
- g. Evaluation of the developed 3D medical visualization system (JSDV) using publicly available real medical datasets for validation based on specific task scenarios. The purpose of the evaluation was to provide answers regarding user performance and user-satisfaction levels of the tested medical students and professionals, validating the research in the real world activities.
- h. Testing the resulting GLmol LEAP application using publicly available 3D molecular models for assessing specific usability metrics based on the defined test cases. The scope of the usability testing was to measure both user performance and user-satisfaction levels of the tested user groups.



As a direct result of the previously exposed objectives the author of the thesis identified the following contributions:

- a. A broad overview of the latest hardware technologies available for future use by various research groups in the development of some novel multimodal user interfaces. Besides the gesture-tracking devices that are providing a natural way of interaction with the displayed 3D models by an autostereoscopic device, the researchers need to consider the opportunities offered by current brain-computer interfaces and tactile devices.
- b. Building of a pseudo-holographic 3D display used by the author of the thesis for testing the applications developed during the research phase. This custom-made 3D autostereoscopic device allowed the test participants to have a more realistic alternative representation of the visualized 3D models providing a better understanding of the studied images. This low-cost alternative to other types of pseudo-holographic 3D displays had the advantage of a reduced weight, making it appropriate for transportation across the multiple testing locations. It also served as a practical tool for explaining the concept and working principles of an autostereoscopic 3D display to the users.
- c. A comprehensive analysis of the current software technologies used during diverse phases of the research. The presented open source libraries and APIs are constituting a valuable tool for every researcher faced with the development of viable software alternatives (as demonstrated by the thesis) that do not depend on specific pre-installed operating systems or proprietary software components. The investigated software solutions are applicable to other development areas where gesture tracking could be used to understand or interact with the studied 3D models (e.g. 3D game development, 3D animations in movies, architecture or various engineering fields). The extra calibration steps needed in setups where multiple cameras are used for gesture tracking, are eliminated by using the LeapJS library in conjunction with a LEAP Motion device. This reduces the times needed for user switching to none when they are commonly interacting with a rendered 3D model. The cross-platform software solutions introduced in chapter 3 provide a solid base for extending the applications if needed by adding functionalities from other devices (e.g. head-mounted displays).
- d. A client-side DICOM viewer named JSDV. This browser-based medical imaging application provided both a regular 2D user interface and a novel 3D interface that allowed the users to interact with the rendered 3D models based on patients' DICOM files with the help of diverse hand gestures.
- e. The development of a new segmentation algorithm with a variable threshold that significantly reduces the noise (that is usually present in the edges) resulting in an adjustable smooth contour of the extracted images. The algorithm improves the execution speed since it checks only one neighbouring pixel (instead of three or four) and was needed due to the maximum execution time limitations imposed in a browser-based environment mentioned in chapter 4. This segmentation algorithm provides a very useful alternative in other areas where the fast detection of specific features inside the analyzed images is critical such as machine vision,

- content-based image retrieval, object detection and tracking across multiple video frames.
- f. The implementation of an interactive client-side colorization algorithm that has been used for highlighting specific tissue types inside the rendered 2D images that provided a base for the multiplanar 3D reconstruction of the studied anatomical area of the patient.
  - g. Updating the ThreeLeapControls JavaScript library to the latest functionalities provided by the current LeapJS v 0.6.4 and ThreeJS rev. 79. The new version detects the number of individual fingers that are extended and provides gesture-based interaction with the rendered 3D models inside both JSDV and GLmol LEAP applications.
  - h. Adapting a client-side 3D molecular viewer to gesture-based interaction. The newly implemented gesture tracking functionalities were possible due to the fact that GLmol uses ThreeJS for rendering that allowed the interfacing of LeapJS using the modified ThreeLeapControls library. The new version named by the author of the thesis GLmol LEAP allows the users to interact with the rendered 3D chemical structures in its standard way (with a 2D mouse) and with the help of a LEAP Motion device if available providing extended multimodal user interaction capabilities.
  - i. Proposing a sound methodology and testing of the JSDV application with the help of specific user groups. The author of the thesis used publicly available DICOM files for the evaluation of the newly developed 3D medical visualization system. The results were analyzed in chapter 6 in order to provide insights from the user performance and user satisfaction point of view. The forms used during testing and the gathered raw data are available in the Annex 1 of the thesis.
  - j. The evaluation of the extended GLmol application (GLmol LEAP) has been done also with the help of specific user groups by using publicly available 3D macromolecular models. The assessed usability metrics provided answers regarding user performance and satisfaction levels. The testing results are exposed and analyzed in chapter 6 of the thesis. The forms used during evaluation and the collected raw data are available in the Annex 2 of the thesis.

The thesis results add formal and practical knowledge in terms of methodology, algorithms, implementation, and evaluation regarding the various hardware and software technologies that are representing a key component in the development of applications that are used to manipulate pseudo-holographic images using gestures. The complexity of the subject implies further work that might be based on the authors current findings.

Further development may include the extension of the applications for sound-based gesture detection as presented in chapter 2. The SoundWave technique could eliminate the need of using a LEAP Motion controller for gesture detection making the applications even more affordable from the hardware point of view. The AirLink system could be used to share the data between multiple mobile devices.

Inclusion of tactile feedback will also be considered in future versions of the applications extending the range of the senses involved in the user interaction. This feature might be useful in the development of a virtual operating room that might be remotely controlled with the help of WebRTC.

The building of a larger pseudo-holographic display is also considered by the author of the thesis for telepresence applications used either by remote training or as a conferencing tool.

The solutions presented through the thesis provide a strong base for the development of future virtual laboratories in any discipline that can benefit from the gesture-based interaction with the rendered 3D pseudo-holographic images.

## **ANNEX 1**

This annex contains the annexes of the JSDV usability test and raw test results data.

Figure A.1 shows the annex 1 - Consent form for the JSDV application.

### **Consent Form (Adult)**

I agree to participate in the study conducted by the Politehnica University of Timisoara regarding the JSDV (JavaScript DICOM Viewer) application.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

**Date:** \_\_\_\_\_

**Please print your name:** \_\_\_\_\_

**Please sign your name:** \_\_\_\_\_

**Thank you!**

We appreciate your participation.

Figure A.1. Consent form for the JSDV application

Figure A.2 shows the annex 2 - Demographic information form for the JSDV application.

## JSDV (JavaScript DICOM Viewer) Usability Test Participant Personal Information

The following information you provide is kept completely confidential and will be used only for the demographical analysis of the test participants. Please fill in the following fields according to your status:

Name.....

Age .....

Gender    male     female

Professional status    physician                       medical student

Clinical experience (years if applicable).....

Specialty (if applicable).....

Your previous experience with computers (years of using one if applicable).....

Previous experience with medical software (years if applicable).....

Previous experience with 2D medical datasets (years if applicable).....

Previous experience with 3D medical datasets (years if applicable).....

Are you comfortable using a web browser?                      yes                       no

Are you comfortable using a 2D mouse?                      yes                       no

Previous experience with a LEAP Motion Device                      yes                       no

Are you left or right handed?                      left                       right

Previous experience with a 3D holographic display                      yes                       no

Other relevant information (if applicable).....

Figure A.2. Demographic information form for the JSDV application

Figure A.3 shows the annex 3 - Task Scenarios for the JSDV application.

Task Scenarios for JSDV (JavaScript DICOM Viewer)

Task No.	Task description	Required to perform	Successful completion criteria	Maximum time
1	Find out how to start using the JSDV application.	<ul style="list-style-type: none"> <li>click the "Help" button</li> <li>click "Select a DICOM file" button</li> </ul>	Participant clicked the "Help" button and after that the "Select a DICOM file" button (this confirms that he read the help content).	120 s
2	Locate and open a DICOM file for segmentation.	<ul style="list-style-type: none"> <li>click "Select a DICOM file" button</li> <li>select a DICOM file from the local file system</li> </ul>	Participant clicked on the "Select a DICOM file" button, located the file on the computer and opened it.	60 s
3	Obtain a well defined contour of the image loaded from the DICOM file.	<ul style="list-style-type: none"> <li>adjust the "Segmentation threshold" cursor in the user interface</li> </ul>	User managed to isolate the most relevant part of the area of interest.	120 s
4	Isolate a tissue type in the colorized image.	<ul style="list-style-type: none"> <li>adjust the min and max levels of the colorization module</li> </ul>	User managed to isolate a tissue based on its color.	120 s
5	If satisfied with the colorization result, save the resulting image for the 3D reconstruction.	<ul style="list-style-type: none"> <li>click the "Save image" button and save the file as PNG.</li> </ul>	User clicked the "Save image" button.	60 s
6	Load a set of images for the 3D reconstruction.	<ul style="list-style-type: none"> <li>click the "Select saved images" button</li> <li>select all the PNG files from the application's img folder</li> <li>open the images</li> </ul>	Participant clicked on the "Select saved images" button, selected the local images and opened them.	60 s
7	Rotate the rendered 3D model with the mouse.	<ul style="list-style-type: none"> <li>click and hold the left mouse button with the cursor on the rendered 3D model</li> <li>drag to rotate</li> </ul>	User clicked on the 3D model and rotated it around the y axis.	60 s
8	Select and isolate a specific slice from the 3D stack with the help of the LEAP device.	<ul style="list-style-type: none"> <li>user points towards a slice with the index finger and when it changes its opacity he uses both hands (10 fingers extended) to move aside the slice.</li> </ul>	User pointed towards the 3D model (1 hand and 1 finger) and used both hands to put aside a slice from the 3D stack.	120 s

Figure A.3. Task Scenarios for the JSDV application

Figure A.4 shows the annex 4 - Time on task table for the JSDV application.

**Time on Task table for JSDV (JavaScript DICOM Viewer)**

	TASK SCENARIOS								Total Time / User
	1	2	3	4	5	6	7	8	
Max allowed time	120 s	60 s	120 s	120 s	60 s	60 s	60 s	120 s	120 s
User 1									
User 2									
User 3									
User 4									
User 5									
User 6									
User 7									
User 8									
User 9									
User 10									
Average times									

Figure A.4. Time on task table for the JSDV application

Figure A.5 shows the annex 5 - Usability Questionnaire for the JSDV application.

Your Name.....

Please answer all the questions by putting an x next to your choice.

		YES	yes	uncertain	no	NO
1	I feel comfortable using this software.					
2	The information presented on the screen is clear enough to understand how to operate the JSDV application.					
3	I think the software responds too slowly at certain stages.					
4	I have enough information to operate the application.					
5	I think the user interface is organized in a simple and consistent manner.					
6	Learning to operate the software was too hard.					
7	I think that the help section is not very useful.					
8	I would like to use this software at my workplace.					
9	It was easy to find out what to do next when I got stuck.					
10	The response time of the software is really fast.					
11	There is no relevant information on the screen in order to operate the software.					
12	There is too much information that I have to understand before I can operate the application.					
13	The way the buttons are organized is logical and relevant.					
14	I would recommend this software to my colleagues.					

Figure A.5. Usability Questionnaire for the JSDV application



Figure A.6 shows the annex 6 - 2D Mouse Device Assessment Questionnaire for the JSDV application.

## 2D Mouse Device Assessment Questionnaire

**Your Name** .....

Please circle the **x** that is most appropriate as an answer to each comment below.

1. The operation speed of the mouse was  

x	x	x	x	x
too fast				too slow
  
2. Finger fatigue due to operating the device was  

x	x	x	x	x
none				very high
  
3. Accurate pointing on the screen was  

x	x	x	x	x
easy				difficult
  
4. The force required for actuation was  

x	x	x	x	x
too low				too high
  
5. The mental effort required for operation was  

x	x	x	x	x
too low				too high
  
6. The physical effort required for operation was  

x	x	x	x	x
too low				too high
  
7. Smoothness during operation was  

x	x	x	x	x
very smooth				very rough
  
8. Wrist fatigue during operation was  

x	x	x	x	x
none				very high

9. Arm fatigue due to the operation was

x	x	x	x	x
very high				none

10. Shoulder fatigue due to operation was

x	x	x	x	x
none				very high

11. Neck fatigue due to operation was

x	x	x	x	x
very high				none

12. General comfort

x	x	x	x	x
very uncomfortable				very comfortable

13. Overall, the 2D mouse was

x	x	x	x	x
very easy to use				very difficult to use

Figure A.6. 2D Mouse Device Assessment Questionnaire for the JSDV application

Figure A.7 shows the annex 7 - LEAP Device Assessment Questionnaire for the JSDV application.

## LEAP Motion Device Assessment Questionnaire

**Your Name**.....

Please circle the **x** that is most appropriate as an answer to each comment bellow.

1. The operation speed of the LEAP Motion Controller was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
too fast				too slow

2. Finger fatigue due to operating the LEAP device was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
none				very high

3. Accurate pointing on the screen was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
easy				difficult

4. The force required for actuation was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
too low				too high

5. The mental effort required for operation was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
too low				too high

6. The physical effort required for operation was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
too low				too high

7. Smoothness during operation was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
very smooth				very rough

8. Wrist fatigue during operation was

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
none				very high

9. Arm fatigue due to the operation was

x	x	x	x	x
very high				none

10. Shoulder fatigue due to operation was

x	x	x	x	x
none				very high

11. Neck fatigue due to operation was

x	x	x	x	x
very high				none

12. General comfort

x	x	x	x	x
very uncomfortable				very comfortable

13. Overall, the LEAP Motion Controller was

x	x	x	x	x
very easy to use				very difficult to use

Figure A.7. LEAP Device Assessment Assessment Questionnaire for the JSDV application

Figure A.8 shows the raw test results for annex 2 - for the JSDV application.

User	Age	Gender	Professional status	Clinical experience (Years)	Specialty	Previous experience with computers (years)	Previous experience with medical software (years)	Previous experience with 2D medical datasets (years)	Previous experience with 3D medical datasets (years)	Are you comfortable using a web browser?	Are you comfortable using a 2D mouse?	Previous experience with a LEAP Motion Device?	Are you left or right handed?	Previous experience with a 3D holographic display?
u 1	22	F	physician	0.5	Medical rehabilitation	10				yes	yes	no	right	no
u 2	22	F	medical student		Medical rehabilitation	10				yes	yes	no	right	no
u 3	22	F	medical student	0.25	Medical rehabilitation	10	3			yes	yes	no	right	no
u 4	24	M	medical student	4	Medical Engineering	10	4			yes	yes	no	right	no
u 5	23	M	master student		Mechanical Engineering	10	1	5	1	yes	yes	yes	right	yes
u 6	23	M	master student		Mechanical Engineering	16				yes	yes	no	right	no
u 7	23	M	master student		Mechanical Engineering	14				yes	yes	no	right	no
u 8	26	M	PHD student		IT & C	8	3	3	2	yes	yes	yes	right	yes
u 9	20	F	medical student			15				yes	yes	no	right	no
u 10	19	F	medical student			13				yes	yes	no	right	yes
u 11	19	F	medical student			8				yes	yes	no	right	no
u 12	19	F	medical student			7				yes	yes	no	right	no
u 13	19	F	medical student			7				yes	yes	no	right	no
u 14	19	F	medical student			7				yes	yes	no	right	no
u 15	20	F	medical student			13				yes	yes	no	right	no
u 16	20	F	medical student			6				yes	yes	no	right	no
u 17	20	F	medical student			6				yes	yes	no	right	no
u 18	19	F	medical student			10				yes	yes	no	right	no
u 19	19	M	medical student			9				yes	yes	no	right	no
u 20	19	M	medical student			10				yes	yes	no	right	no
u 21	20	M	medical student			7				yes	yes	no	right	no
u 22	21	F	medical student			10				yes	no	no	right	no
u 23	19	F	medical student			10				yes	yes	no	right	no
u 24	19	F	medical student			12				yes	yes	no	right	no
u 25	19	F	medical student			11				yes	yes	no	right	no
u 26	19	F	medical student			10				yes	yes	no	right	no
u 27	20	M	medical student			12				yes	yes	no	right	no
u 28	19	M	medical student			10				yes	yes	no	right	no
u 29	19	F	medical student			12				yes	yes	no	right	no
u 30	19	F	medical student			10				yes	yes	no	right	no
u 31	18	F	medical student			13				yes	yes	no	right	no
u 32	20	F	medical student			10				yes	yes	no	right	no

Figure A.8. Raw test results for annex 2 for the JSDV application

Figure A.9 shows the raw test results for annex 5 - for the JSDV application.

SCALE	Metrics short notation	NO = 1	no = 2	uncertain = 3	yes = 4	YES = 5	L	HS	CA	HS	SR	R	L	R	CA
		user control and freedom	consistency	software response speed	user control and freedom	consistency	learnability	help section related issues	clinical applicability	help section related issues	software response speed	relevance	learnability	relevance	clinical applicability
	<b>Questions</b>	The information presented on the screen is comfortable using this software.	I feel clear enough to understand how to operate the software.	I think the JSDV application.	I have enough information to operate the application.	I think the user interface is organized in a simple and consistent manner.	Learning to operate the software was too hard.	I think the help section is not very useful.	I would like to use this software at my workplace.	It was easy to find out what to do next when I got stuck.	The response time of the software is fast.	There is no information on the screen in order to operate the software.	There is too much information that I have to understand before I can operate the application.	The way the buttons are organized is logical and relevant.	I would recommend this software to my colleagues.
u1		5	5	2	5	5	2	1	5	4	4	1	2	5	4
u2		5	5	1	3	5	1	1	5	4	4	1	1	5	5
u3		5	3	1	4	5	1	1	5	3	4	2	2	5	5
u4		5	5	1	5	5	1	1	5	5	5	1	1	5	5
u5		4	3	3	4	4	4	2	2	3	4	3	2	4	4
u6		5	5	4	3	5	1	1	3	5	4	1	2	5	5
u7		5	4	2	4	4	2	4	3	3	4	3	2	4	3
u8		5	5	1	5	5	1	1	5	4	4	1	2	5	5
u9		5	5	1	4	5	1	1	5	4	5	1	1	5	5
u10		3	4	4	2	4	3	3	2	3	4	2	3	4	3
u11		5	4	4	3	4	1	2	5	4	4	2	1	5	5
u12		4	5	3	4	5	1	2	5	5	3	2	2	5	5
u13		4	4	3	4	5	2	1	4	3	4	1	2	5	4
u14		5	4	4	5	4	2	1	5	5	5	1	3	5	5
u15		4	5	2	4	5	1	1	5	4	4	1	3	4	5
u16		5	5	1	5	5	2	2	5	5	4	1	2	4	5
u17		5	4	1	5	5	1	2	4	4	1	2	1	5	5
u18		5	5	1	5	2	2	1	5	5	5	1	1	5	5
u19		3	4	2	4	4	3	2	4	2	5	2	4	3	4
u20		4	5	2	5	4	1	1	4	4	4	3	2	5	4
u21		4	5	2	3	4	2	3	4	4	4	3	3	4	5
u22		5	5	4	4	5	1	2	5	5	5	1	1	5	5
u23		5	5	1	5	5	1	1	5	5	5	1	1	5	5
u24		5	5	1	5	5	1	1	5	4	5	1	1	5	5
u25		4	5	4	5	5	1	1	4	4	3	1	1	5	4
u26		5	4	2	5	4	1	1	5	4	4	1	2	4	5
u27		5	4	4	4	4	5	2	4	4	4	4	4	4	5
u28		5	4	1	4	5	1	1	5	5	4	3	2	4	5
u29		5	4	2	4	5	1	2	5	4	3	1	2	4	5
u30		5	5	1	4	5	1	4	4	4	5	2	1	5	4
u31		4	5	1	5	5	1	1	5	5	5	1	1	5	5
u32		4	5	2	5	5	1	1	5	5	4	1	1	5	5

Figure A.9. Raw test results for annex 5 for the JSDV application

Figure A.10 shows the raw test results for annex 6 - for the JSDV application.

Question No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
<b>Questions</b>	The operation speed of the mouse to operate the device was	Finger fatigue due to operating the device was	Accurate pointing on the screen was	The force for actuation was	The mental effort required for operation was	The physical effort required for operation was	Smoothness during operation was	Wrist fatigue during operation was	Arm fatigue due to the operation was	Shoulder fatigue due to the operation was	Neck fatigue due to the operation was	General comfort	Overall, the 2D mouse was
1	too fast	none	easy	too low	too low	too low	very smooth	none	very high	none	very high	very uncomfortable	very easy to use
2													
3													
4													
5	too slow	very high	difficult	too high	too high	too high	very rough	very high	none	very high	none	very comfortable	very difficult to use
u1	2	1	1	3	2	1	3	1	5	1	5	4	1
u2	1	1	1	5	1	1	1	1	5	1	5	5	1
u3	3	1	1	3	3	3	1	1	5	1	5	5	1
u4	1	5	1	3	3	2	1	3	3	3	3	5	1
u5	3	4	2	3	3	3	2	2	2	3	3	4	4
u6	4	1	2	3	2	2	2	1	5	1	5	4	2
u7	3	1	1	3	2	2	1	1	5	1	5	4	1
u8	1	4	2	2	2	2	2	2	4	2	4	4	1
u9	2	1	1	2	2	1	1	1	5	1	5	5	1
u10	2	4	2	2	2	2	3	2	4	2	5	4	2
u11	2	2	3	2	1	1	2	2	4	2	5	4	1
u12	2	4	1	4	4	4	3	3	4	3	5	5	1
u13	4	1	1	2	4	2	1	1	5	1	5	4	3
u14	2	3	1	3	4	5	3	3	3	1	5	4	1
u15	2	1	3	2	2	2	1	1	5	1	5	5	1
u16	2	2	2	1	3	1	2	1	5	4	4	4	2
u17	3	3	1	2	3	3	3	3	3	3	3	4	1
u18	1	5	1	1	3	2	1	3	4	1	4	5	1
u19	3	1	1	3	2	2	2	1	5	1	5	1	1
u20	3	2	1	2	2	1	2	1	5	1	4	4	2
u21	2	4	3	4	2	2	2	2	4	1	5	3	2
u22	1	1	1	1	3	1	1	1	5	1	5	5	1
u23	2	1	1	1	2	2	1	1	5	1	5	5	1
u24	3	1	2	2	1	1	2	2	5	1	5	4	1
u25	3	1	3	1	2	2	1	1	5	1	5	4	1
u26	3	1	2	3	3	2	3	1	5	1	5	1	2
u27	1	5	1	1	1	5	1	5	5	1	5	5	1
u28	2	1	1	3	2	2	1	1	5	1	5	4	2
u29	3	1	2	3	2	2	3	1	4	1	5	5	1
u30	2	1	1	2	2	2	1	1	5	1	5	1	1
u31	2	3	1	3	3	3	3	1	5	1	5	4	1
u32	3	1	2	3	3	3	3	1	5	1	5	5	1

Figure A.10. Raw test results for annex 6 for the JSDV application

Figure A.11 shows the raw test results for annex 7 - for the JSDV application.

Question No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
Questions	The operation speed of the LEAP Motion Controller was	Finger fatigue due to operating the LEAP device was	Accurate pointing on the screen was	The force for actuation was	The mental effort required for operation was	The physical effort required for operation was	Smoothness during operation was	Wrist fatigue during operation was	Arm fatigue due to the operation was	Shoulder fatigue due to the operation was	Neck fatigue due to the operation was	General comfort	Overall, the LEAP Motion Controller was
1	too fast	none	easy	too low	too low	too low	very smooth	none	very high	none	very high	very uncomfortable	very easy to use
2													
3													
4													
5	too slow	very high	difficult	too high	too high	too high	very rough	very high	none	very high	none	very comfortable	very difficult to use
u1	3	1	2	1	2	1	3	1	5	1	5	4	2
u2	1	1	1	5	1	1	1	1	5	1	5	5	1
u3	3	1	3	3	3	3	1	1	5	1	5	5	3
u4	1	5	1	4	1	1	1	3	3	3	3	5	1
u5	4	4	3	4	2	2	2	3	3	2	3	4	3
u6	3	1	2	2	2	1	2	1	5	1	5	2	2
u7	4	2	4	2	4	2	4	2	2	4	4	3	4
u8	1	4	2	3	2	2	2	2	4	2	4	4	2
u9	2	2	2	2	3	2	1	2	4	1	5	5	2
u10	4	2	1	2	2	2	3	2	4	2	5	4	3
u11	2	2	2	1	2	1	2	2	4	2	5	4	2
u12	3	3	4	3	2	2	3	2	2	2	2	5	3
u13	2	1	1	2	1	1	1	1	5	1	5	5	1
u14	2	4	3	4	4	3	3	3	3	3	3	4	2
u15	2	1	3	2	3	2	1	1	5	1	5	5	2
u16	2	1	2	1	3	1	2	1	4	1	4	4	2
u17	4	3	3	3	3	3	3	3	3	3	3	4	3
u18	2	1	1	1	3	2	1	1	5	1	5	5	2
u19	4	4	2	4	3	3	2	3	3	1	5	3	3
u20	2	2	4	3	3	2	4	2	4	2	4	4	3
u21	3	3	3	2	3	2	1	2	4	2	5	3	2
u22	2	1	2	2	2	2	2	1	5	1	5	5	1
u23	2	1	1	2	1	2	1	1	5	1	5	5	1
u24	3	1	1	1	1	2	1	2	4	4	5	4	1
u25	3	2	3	3	2	2	2	1	4	1	5	3	2
u26	3	1	2	2	3	2	3	1	5	1	5	4	3
u27	5	1	1	1	1	1	1	5	5	5	1	5	1
u28	2	1	2	2	2	2	2	1	5	1	5	4	2
u29	3	1	2	3	2	2	2	1	4	4	5	5	1
u30	2	1	1	2	3	2	1	1	5	1	5	5	1
u31	3	1	4	3	3	3	3	1	5	1	5	5	1
u32	3	1	2	2	2	2	2	1	5	1	5	5	1

Figure A.11. Raw test results for annex 7 for the JSDV application



## ANNEX 2

This annex contains the annexes of the GLmol LEAP usability test and raw test results data.

Figure A2.1 shows the annex 1 - Consent form for the GLmol LEAP application.

### Consent Form (Adult)

I agree to participate in the study conducted by the Politehnica University of Timisoara regarding the GLmol LEAP (GLmol Molecular Viewer with LEAP gesture control) application.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

**Date:** \_\_\_\_\_

**Please print your name:** \_\_\_\_\_

**Please sign your name:** \_\_\_\_\_

**Thank you!**

We appreciate your participation.

Figure A2.1. Consent form for the GLmol LEAP application

Figure A2.2 shows the annex 2 - Demographic information form for the GLmol LEAP application.

### GLmol LEAP (GLmol Molecular Viewer with LEAP gesture control) Usability Test Participant Personal Information

The following information you provide is kept completely confidential and will be used only for the demographical analysis of the test participants. Please fill in the following fields according to your status:

Name.....

Age .....

Gender male  female

Professional status chemistry student  other

Specialty (if applicable).....

Your previous experience with computers (years of using one if applicable).....

Previous experience with 3D molecular viewers (years if applicable).....

Previous experience with 3D molecular datasets (years if applicable).....

Are you comfortable using a web browser? yes  no

Are you comfortable using a 2D mouse? yes  no

Previous experience with a LEAP Motion Device yes  no

Are you left or right handed? left  right

Previous experience with a 3D holographic display yes  no

Other relevant information (if applicable).....

Figure A2.2. Demographic information form for the GLmol LEAP application

Figure A2.3 shows the annex 3 - Task Scenarios for the GLmol LEAP application.

Task Scenarios for GLmol LEAP (GLmol Molecular Viewer with LEAP gesture control)

Task No.	Task description	Required to perform	Successful completion criteria	Maximum time
1	Find out how to start using the GLmol LEAP application.	<ul style="list-style-type: none"> <li>click the "About" button</li> <li>click on the rendered 3D model and rotate it (or use gestures to rotate it)</li> </ul>	Participant clicked the "About" button and after that clicked on the 3D model and rotated it with the mouse or with the help of the LEAP Motion Controller (this confirms that he read the help content).	60 s
2	Locate and open a local PDB file for visualization.	<ul style="list-style-type: none"> <li>click the "Load" button</li> <li>click the "Browse" button, select a PDB file from the local file system</li> <li>click the "Load from file" button</li> </ul>	Participant clicked on the "Load" button, located the file on the computer, opened and loaded it.	60 s
3	Rotate the rendered 3D model with the mouse.	<ul style="list-style-type: none"> <li>click and hold the left mouse button with the cursor on the rendered 3D model</li> <li>drag to rotate</li> </ul>	User clicked on the 3D model and rotated it around the y axis.	30 s
4	Enlarge the 3D model with the help of the LEAP device.	<ul style="list-style-type: none"> <li>user uses one hand (5 fingers extended) to zoom in and out.</li> </ul>	User extended his arm towards the 3D model (1 hand and 5 fingers) and managed to enlarge the rendered 3D model.	30 s

Figure A2.3. Task Scenarios for the GLmol LEAP application

Figure A2.4 shows the annex 4 - Time on task table for the GLmol LEAP application.

**Time on Task table for GLmol LEAP** (GLmol Molecular Viewer with LEAP gesture control)

	TASK SCENARIOS				Total Time / User
	1	2	3	4	
Max allowed time	60 s	60 s	30 s	30 s	
User 1					
User 2					
User 3					
User 4					
User 5					
User 6					
User 7					
User 8					
User 9					
User 10					
Average times					

Figure A2.4. Time on task table for the GLmol LEAP application

Figure A2.5 shows the annex 5 - Usability Questionnaire for the GLmol LEAP application.

Your Name .....

Please answer all the questions by putting an x next to your choice.

		YES	yes	uncertain	no	NO
1	I feel comfortable using this software .					
2	The information presented on the screen is clear enough to understand how to operate the GLmol LEAP application.					
3	I think the software responds too slowly at certain stages.					
4	I have enough information to operate the application.					
5	I think the user interface is organized in a simple and consistent manner.					
6	Learning to operate the software was too hard.					
7	I think that the help section is not very useful.					
8	I would like to use this software at my workplace.					
9	It was easy to find out what to do next when I got stuck.					
10	The response time of the software is really fast.					
11	There is no relevant information on the screen in order to operate the software .					
12	There is too much information that I have to understand before I can operate the application.					
13	The way the buttons are organized is logical and relevant.					
14	I would recommend this software to my colleagues.					

Figure A2.5. Usability Questionnaire for the GLmol LEAP application

Figure A2.6 shows the annex 6 - 2D Mouse Device Assessment Questionnaire for the GLmol LEAP application.

## 2D Mouse Device Assessment Questionnaire

**Your Name**.....

Please circle the **x** that is most appropriate as an answer to each comment bellow.

1. The operation speed of the mouse was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
too fast				too slow
  
2. Finger fatigue due to operating the device was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
none				very high
  
3. Accurate pointing on the screen was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
easy				difficult
  
4. The force required for actuation was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
too low				too high
  
5. The mental effort required for operation was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
too low				too high
  
6. The physical effort required for operation was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
too low				too high
  
7. Smoothness during operation was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
very smooth				very rough
  
8. Wrist fatigue during operation was  

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
none				very high

9. Arm fatigue due to the operation was

x	x	x	x	x
very high				none

10. Shoulder fatigue due to operation was

x	x	x	x	x
none				very high

11. Neck fatigue due to operation was

x	x	x	x	x
very high				none

12. General comfort

x	x	x	x	x
very uncomfortable				very comfortable

13. Overall, the 2D mouse was

x	x	x	x	x
very easy to use				very difficult to use

Figure A2.6. 2D Mouse Device Assessment Questionnaire for the GLmol LEAP application





9. Arm fatigue due to the operation was

x x x x x  
very high none

10. Shoulder fatigue due to operation was

x x x x x  
none very high

11. Neck fatigue due to operation was

x x x x x  
very high none

12. General comfort

x x x x x  
very very  
uncomfortable comfortable

13. Overall, the LEAP Motion Controller was

x x x x x  
very easy very difficult  
to use to use

Figure A2.7. LEAP Device Assessment Questionnaire for the GLmol LEAP application

Figure A2.8 shows the raw test results for annex 2 - for the GLmol LEAP application.

User	Age	Gender	Professional status	Specialty	Previous experience with computers (years)	Previous experience with 3D molecular viewers (years)	Previous experience with 3D molecular datasets (years)	Are you comfortable using a web browser?	Are you comfortable using a 2D mouse?	Previous experience with a LEAP Motion Device?	Are you left or right handed?	Previous experience with a 3D holographic display?
u 1	48	F	assistant professor	Organic chemistry	20			no	yes	no	right	no
u 2	30	F	assistant professor	Organic chemistry	10	6	6	yes	yes	no	right	no
u 3	36	F	assistant professor	Organic chemistry	22	9	9	yes	yes	no	right	no
u 4	48	F	lecturer	Organic chemistry	20			yes	yes	no	left	no
u 5	47	M	Associate professor	Organic chemistry	22	20	20	yes	yes	no	right	no
u 6	36	F	assistant professor	Organic chemistry	17	5	5	yes	yes	no	right	no
u 7	33	F	assistant professor	Organic chemistry	17	5	3	yes	no	no	right	no
u 8	43	F	lecturer	Organic chemistry	20			yes	yes	no	right	no
u 9	32	F	assistant professor	Organic chemistry	20	9	9	yes	yes	no	right	no
u 10	49	M	lecturer	Organic chemistry	20	8		yes	yes	no	right	no
u 11	57	M	Associate professor	Organic chemistry	30	20	20	yes	yes	no	right	no
u 12	26	M	PhD student		8			yes	yes	yes	right	yes
u 13	21	F	student	chemistry	14			yes	yes	no	right	no
u 14	20	M	student	chemistry	10			yes	yes	no	right	no
u 15	20	F	student	chemistry	10	1		yes	yes	no	right	no
u 16	21	F	student	chemistry	10	1		yes	yes	no	right	no
u 17	21	M	student	chemistry	13	1		yes	yes	no	right	no
u 18	22	F	student	chemistry	11			yes	yes	no	right	yes
u 19	21	F	student	chemistry	13			yes	yes	no	right	yes
u 20	21	F	student	chemistry	14			yes	yes	no	right	yes
u 21	22	F	student	chemistry	10	2		no	yes	no	right	no
u 22	23	F	student	chemistry	10	2		yes	yes	no	right	no
u 23	21	M	student	chemistry	10			yes	yes	no	right	yes

Figure A2.8. Raw test results for annex 2 for the GLmol LEAP application

Figure A2.9 shows the raw test results for annex 5 - for the GLmol LEAP application.

SCALE	Metrics short notation	NO = 1	no = 2	uncertain = 3	yes = 4	YES = 5	L	HS	DA	HS	SR	R	L	R	DA
		UC	C	SR	UC	C	learnability	help section related issues	domain applicability	help section related issues	software response speed	relevance	learnability	relevance	domain applicability
	<b>Questions</b>	I feel comfortable using this software.	The information presented on the screen is clear enough to understand how to operate the GLmol LEAP application.	I think the software responds too slowly at certain stages.	I have enough information to operate the application.	I think the user interface is organized in a simple and consistent manner.	Learning to operate the software was hard.	I think that the help section is not very useful.	I would like to use this software at my workplace.	It was easy to find what to do next when I got stuck.	The response time of the software is really fast.	There is no relevant information on the screen in order to operate the software.	There is too much information that I have to understand before I can operate the application.	The way the buttons are organized is logical and relevant.	I would recommend this software to my colleagues.
u1		5	5	1	2	5	1	2	5	3	5	1	1	5	5
u2		5	4	4	5	5	1	1	5	4	4	1	1	4	5
u3		5	5	2	5	5	1	1	5	5	5	1	1	5	5
u4		5	4	2	3	4	2	2	3	4	5	2	4	4	5
u5		4	4	3	5	5	3	3	3	4	4	4	3	4	4
u6		5	5	1	5	5	1	1	5	5	5	1	1	5	5
u7		5	5	1	5	5	1	2	5	4	4	1	1	4	5
u8		4	4	2	4	4	2	2	5	4	5	2	2	4	4
u9		5	5	2	5	5	1	1	5	4	5	1	1	5	5
u10		4	4	2	5	4	2	2	4	4	4	2	2	4	4
u11		5	5	4	4	5	1	2	4	3	4	1	2	5	5
u12		5	5	1	5	5	1	1	5	5	5	1	1	5	5
u13		4	5	2	2	4	2	4	4	4	4	2	4	4	4
u14		5	5	4	5	5	1	1	5	5	4	1	1	5	5
u15		5	5	2	5	5	1	1	4	5	4	1	1	5	5
u16		5	5	2	5	5	1	1	5	5	5	1	1	5	5
u17		5	5	2	3	4	1	1	4	4	4	2	1	4	4
u18		4	5	2	5	5	1	1	5	5	4	5	1	4	5
u19		4	5	2	5	5	1	1	5	5	4	5	1	4	5
u20		5	5	1	5	5	1	1	5	5	5	1	1	5	5
u21		5	5	3	4	4	1	2	4	4	4	2	1	5	4
u22		5	5	4	5	5	1	1	5	5	4	1	1	5	5
u23		4	4	2	2	4	2	4	5	4	4	4	4	4	4

Figure A2.9. Raw test results for annex 5 for the GLmol LEAP application

Figure A2.10 shows the raw test results for annex 6 - for the GLmol LEAP application.

Question No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
Questions	The operation speed of the mouse was	Finger fatigue due to operating the device was	Accurate pointing on the screen was	The required actuation was	The effort required for operation was	The mental effort required for operation was	The physical effort required for operation was	Wrist fatigue during operation was	Arm fatigue due to operation was	Shoulder fatigue due to operation was	Neck fatigue due to operation was	General comfort	Overall, the 2D mouse was
1	too fast	none	easy	too low	too low	too low	very smooth	none	very high	none	very high	very uncomfortable	very easy to use
2													
3													
4													
5	too slow	very high	difficult	too high	too high	too high	very rough	very high	none	very high	none	very comfortable	very difficult to use
u1	2	1	1	3	2	1	2	1	5	1	5	5	1
u2	3	1	1	2	3	2	2	2	5	1	5	5	2
u3	2	3	2	4	2	3	2	1	5	1	5	5	1
u4	2	2	1	2	4	2	2	2	4	2	4	4	2
u5	3	3	3	3	2	2	3	2	4	2	4	4	2
u6	2	1	1	2	2	2	1	1	5	1	5	5	1
u7	2	1	1	2	2	2	1	1	5	1	5	5	1
u8	3	4	4	3	2	2	2	2	5	2	2	5	1
u9	1	1	1	2	2	1	1	1	5	1	5	5	1
u10	2	4	2	3	2	2	2	3	4	2	4	4	2
u11	2	1	3	2	2	2	2	1	5	1	5	5	1
u12	1	4	1	4	2	2	2	2	4	2	4	5	1
u13	2	3	2	4	2	2	4	1	4	1	5	4	1
u14	3	1	1	1	1	1	2	1	5	1	5	5	1
u15	3	1	1	3	3	3	4	3	3	3	3	1	2
u16	3	3	2	3	2	2	3	3	3	2	1	5	1
u17	2	3	2	4	2	2	3	3	3	2	5	4	1
u18	3	1	2	3	3	3	4	1	1	1	1	4	1
u19	3	1	2	2	2	1	4	1	1	1	1	5	1
u20	2	1	2	3	2	2	2	1	5	1	5	5	1
u21	3	3	2	3	1	2	3	2	4	2	4	5	1
u22	2	1	1	3	2	2	2	1	5	1	5	5	1
u23	2	2	2	3	2	1	4	1	4	2	5	5	1

Figure A2.10. Raw test results for annex 6 for the GLmol LEAP application

Figure A2.11 shows the raw test results for annex 7 - for the GLmol LEAP application.

Question No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13
Questions	The operation speed of the LEAP Motion Controller was	Finger fatigue due to operating the LEAP device was	Accurate interaction with the 3D model rendered on the screen was	The force required for the actuation was	The mental effort required for operation was	The physical effort required for operation was	Smoothness during operation was	Wrist fatigue during operation was	Arm fatigue due to operation was	Shoulder fatigue due to operation was	Neck fatigue due to operation was	General comfort	Overall, the LEAP Motion Controller was
1	too fast	none	easy	too low	too low	too low	very smooth	none	very high	none	very high	very uncomfortable	very easy to use
2													
3													
4													
5	too slow	very high	difficult	too high	too high	too high	very rough	very high	none	very high	none	very comfortable	very difficult to use
u1	2	1	1	1	1	1	2	1	5	1	5	5	1
u2	2	2	3	4	2	2	2	2	5	1	5	5	2
u3	2	4	1	4	1	1	1	1	5	1	5	5	1
u4	3	2	3	2	3	2	2	2	4	4	4	4	2
u5	4	3	4	3	3	2	3	2	4	2	4	4	2
u6	2	1	2	2	2	2	1	1	4	2	4	5	1
u7	2	1	1	2	2	2	1	2	5	1	5	5	1
u8	2	1	1	3	2	2	3	2	4	4	3	5	1
u9	3	1	2	2	2	1	2	1	5	1	5	4	2
u10	2	3	4	3	2	3	3	4	3	3	4	3	3
u11	2	1	2	2	2	1	2	1	5	1	5	5	2
u12	2	4	1	4	2	2	2	2	4	2	4	5	1
u13	3	3	3	3	2	2	4	2	4	1	5	4	1
u14	2	1	1	1	1	1	2	1	5	1	5	5	2
u15	4	3	2	3	3	3	2	3	3	3	3	5	1
u16	2	3	2	2	3	2	3	3	3	2	5	5	1
u17	2	3	2	4	2	2	4	3	4	4	4	5	2
u18	3	1	2	3	3	3	4	1	4	1	5	4	2
u19	3	2	2	2	1	1	4	1	4	1	5	4	2
u20	3	1	1	3	3	3	1	1	5	1	5	5	1
u21	3	2	1	1	1	1	2	2	4	2	4	5	1
u22	2	1	1	2	2	2	2	1	5	1	5	5	1
u23	2	2	3	2	1	1	4	2	3	2	5	5	2

Figure A2.11. Raw test results for annex 7 for the GLmol LEAP application

## REFERENCES

- [1] R. H. Choplin, J. M. Boehme, 2nd, and C. D. Maynard, "Picture archiving and communication systems: an overview," *Radiographics*, vol. 12, no.1, 1992, pp. 127-129
- [2] Digital Imaging and Communications in Medicine, available online at <http://dicom.nema.org/>, last accessed on 09.07.2016.
- [3] E. J. M. Monteiro, C. Costa, and J. L. Oliveira, "A DICOM Viewer based on Web Technology," *IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2013, pp. 167-171
- [4] L.C. Ebert, G. Hatch, M.J. Thali, and S. Ross, "Invisible touch—Control of a DICOM viewer with finger gestures using the Kinect depth camera", *Journal of Forensic Radiology and Imaging*, vol. 1, issue 1, 2013, pp. 10-14
- [5] N. Rossol, I. Cheng, R. Shen and A. Basu, "Touchfree medical interfaces," *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, August 2014, pp. 6597 - 6600
- [6] C. Kirmizibayrak et al., "Evaluation of gesture based interfaces for medical volume visualization tasks," *In Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI '11)*, 2011, pp. 69-74.
- [7] L. Lopez Fernandez, M. Paris Diaz, R. Benitez Mejias, F. J. Lopez, J. A. Santos, "Kurento: a media server technology for convergent WWW/mobile real-time multimedia communications supporting WebRTC", *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2013, pp. 1-6, doi:10.1109/WoWMoM.2013.6583507
- [8] K. R. Saikaew et al., "CT Image Management and Communication Services," *Service Research and Innovation Institute Global Conference (SRII 12)*, 2012, pp. 660-666
- [9] J. Congote et al., "Volume Ray Casting in WebGL," *Computer Graphics*, N. Mukai, ed., InTech, 2012, pp. 157-178
- [10] F. Steinicke, G. Bruder, K. Hinrichs, T. Ropinski, and M. Lopes, "3D User Interfaces for Collaborative Work," *Human Computer Interaction*, I. Pavlidis, ed., InTech, 2008, pp. 279-294.
- [11] S. Dutton, "Getting Started with WebRTC," *HTML5 Rocks*, 23.07.2012, available online at <https://www.html5rocks.com/en/tutorials/webrtc/basics/> , last accessed on 09.02.2017.

- [12] JavaScript Augmented Reality Toolkit homepage, available online at <https://github.com/kig/JSARToolKit>, last accessed on 09.07.2016.
- [13] ThreeJS homepage, available online at <http://threejs.org/>, last accessed on 09.07.2016.
- [14] NanoDICOM homepage, available online at <http://www.nanodicom.org/>, last accessed on 09.07.2016.
- [15] S. Mann, "Intelligent Image Processing," John Wiley and Sons, 2001, ISBN: 0-471-40637-6.
- [16] D. Wigdor, D. Wixon, "Brave NUI World: Designing Natural User Interfaces for Touch and Gesture," Morgan Kaufmann, 2011, ISBN-10: 0123822319.
- [17] GLmol homepage, available online at <https://github.com/biochem-fan/GLmol>, last accessed on 29.06.2016.
- [18] LeapJS homepage, available online at <https://github.com/leapmotion/leapjs>, last accessed on 07.04.2017.
- [19] ThreeLeapControls homepage, available online at <https://github.com/spren9er/threeleapcontrols>, last accessed on 07.04.2017.
- [20] D. Kammer, M. Keck, G. Freitag, M. Wacker, "Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features", *Workshop on Engineering Patterns for Multitouch Interfaces*, 2010, available online at [https://mg.inf.tu-dresden.de/sites/mg.inf.tu-dresden.de/files/2010\\_taxonomy\\_and\\_overview\\_of\\_multi-touch\\_frameworks.pdf](https://mg.inf.tu-dresden.de/sites/mg.inf.tu-dresden.de/files/2010_taxonomy_and_overview_of_multi-touch_frameworks.pdf), last accessed on 10.07.2016
- [21] M. Kaushik, R. Jain, "Gesture Based Interaction NUI: An Overview," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 9 (nr. 12), pp. 633-636, March 2014.
- [22] N. Gal, D. Andrei, D.I. Nemeş, E. Nădăşan and V. Stoicu-Tivadar, "A Kinect based intelligent e-rehabilitation system in physical therapy," *Studies in Health Technology and Informatics*, vol. 210, Digital Healthcare Empowering Europeans, 2015, pp. 489 - 493.
- [23] M. Riduwan, A. H. Basori, F. Mohamed, "Finger-based Gestural Interaction for Exploration of 3D Heart Visualization," *The 9th International Conference on Cognitive Science*, vol. 97, 2013, pp. 684-690
- [24] N. P. John, K. A. S. Pillai, A. R. Anil, "Video interaction through finger tips," *International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2015, pp. 1-4.

- [25] Z. Kalal, K. Mikolajczyk, J. Matas, "Tracking-Learning-Detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 7, 2012, pp. 1409-1422.
- [26] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," *Sensors (Basel)*, vol. 13, no. 5, May 2013, pp. 6380–6393, doi: 10.3390/s130506380
- [27] R. Goodwins, „Windows 7? No arm in it“, available online at <http://www.zdnet.com/article/windows-7-no-arm-in-it/>, last accessed on 11.07.2016.
- [28] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani, "Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions," *Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI 2014)*, 2014, pp. 1063 - 1072.
- [29] A. Colgan, "How Does the Leap Motion Controller Work?," Leap Motion blog, 09.08.2014, available online at <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>, last accessed on 11.07.2016.
- [30] MYO technical specifications, available online at <https://www.myo.com/techspecs>, last accessed on 10.07.2016.
- [31] "APL's Modular Prosthetic Limb Reaches New Levels of Operability," press release of Johns Hopkins University Applied Physics Laboratory, 12.01.2016, available online at <http://www.jhuapl.edu/newscenter/pressreleases/2016/160112.asp>, last accessed on 11.07.2016.
- [32] C. Zhao, K. Chen, T. I. Aumi, S. Patel and M. S. Reynolds, "SideSwipe: detecting in-air gestures around mobile devices using actual GSM signal," *Proceedings of the 27th annual ACM symposium on User interface software and technology (UIST, 2014)*, pp. 527-534
- [33] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the Doppler Effect to Sense Gestures," *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2012)*, pp. 1911-1914.
- [34] Xin Liu, Kang Li, Dong C. Liu, "A Sound-based Gesture Recognition Technology Designed for Mobile Platform", *Journal of Information & Computational Science*, 2015, Volume 12, Issue 3, pp. 985-991
- [35] K. Chen, D. Ashbrook, M. Goel, S. Lee, and S. Patel, "AirLink: sharing files between multiple devices using in-air gestures," *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014)*, pp. 565-569.
- [36] Project Soli homepage, available online at <https://atap.google.com/soli/>, last accessed on 12.07.2016.



- [37] J. J. Vidal, "Toward Direct Brain-Computer Communication", *Annual Review of Biophysics and Bioengineering*, 1973, Vol. 2, pp.157-180, DOI: 10.1146/annurev.bb.02.060173.001105, available online at <http://www.annualreviews.org/doi/abs/10.1146/annurev.bb.02.060173.001105>, last accessed on 12.07.2016.
- [38] J. Pan et al., "Detecting awareness in patients with disorders of consciousness using a hybrid brain-computer interface", *Journal of Neural Engineering*, 2014, vol. 11 (no. 5), DOI: 10.1088/1741-2560/11/5/056007
- [39] O. Gosseries et al., "Disorders of Consciousness: Coma, Vegetative and Minimally Conscious States ", D. Cvetkovic and I. Cosic (eds.), *States of Consciousness*, The Frontiers Collection, Springer-Verlag Berlin Heidelberg, 2011, pp.29-55, DOI: 10.1007/978-3-642-18047-7\_2
- [40] EPOC homepage, available online at <http://emotiv.com/epoc/>, last accessed on 13.07.2016.
- [41] E. A. Pohlmeyer et al., "Toward the Restoration of Hand Use to a Paralyzed Monkey: Brain-Controlled Functional Electrical Stimulation of Forearm Muscles," *PLoS ONE*, 2009, vol. 4(no. 6): e5924, published online 15.06.2009, DOI: 10.1371/journal.pone.0005924
- [42] A. H. Do, P. T. Wang, C. E. King, S. N. Chun and Z. Nenadic, "Brain-computer interface controlled robotic gait orthosis," *Journal of NeuroEngineering and Rehabilitation*, 2013, 10:111, DOI: 10.1186/1743-0003-10-111
- [43] A. Frotzler et al., "High-volume FES-cycling partially reverses bone loss in people with chronic spinal cord injury," in *Bone*, 2008, vol. 43(no. 1), pp. 169-176, DOI: 10.1016/j.bone.2008.03.004
- [44] E. Waltz, "Spinal Stimulation Gets Paralyzed Patients Moving," published online in *IEEE Spectrum*, 24.10.2013, available online at <http://spectrum.ieee.org/biomedical/devices/spinal-stimulation-gets-paralyzed-patients-moving>, last accessed on 16.07.2016.
- [45] S. Harkema et al., "Effect of Epidural stimulation of the lumbosacral spinal cord on voluntary movement, standing, and assisted stepping after motor complete paraplegia: a case study," in *The Lancet*, 2011, vol. 377 (no. 9781), pp. 1938-1947, DOI: 10.1016/S0140-6736(11)60547-3
- [46] M. Ortiz-Catalan, B. Håkansson and R. Brånemark, "An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs," in *Science Translational Medicine* 08 Oct 2014, Vol. 6, Issue 257, pp. 257re6, DOI: 10.1126/scitranslmed.300893
- [47] A. Nebergall et al., "Stable fixation of an osseointegrated implant system for above-the-knee amputees," *Acta Orthopaedica* , Vol. 83, Issue 2, 2012, pp. 121-128, DOI: 10.3109/17453674.2012.678799

- [48] L. R. Hochberg et al., "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, 2012 May 17, vol. 485( issue 7398): 372–375, DOI: 10.1038/nature11076
- [49] S. Mathôt, J-B. Melmi, L. van der Linden, S. Van der Stigchel, "The Mind-Writing Pupil: A Human-Computer Interface Based on Decoding of Covert Attention through Pupillometry," *PLoS ONE*, 11(2): e0148805, 2016, DOI:10.1371/journal.pone.0148805
- [50] J. Geng, "Three-dimensional display technologies", in *Advances in optics and photonics*, Vol. 5, Issue 4, 2013, pp. 456–535, DOI: 10.1364/AOP.5.000456
- [51] D. F. McAllister, "Stereo and 3D Display Technologies", in *Encyclopedia of Imaging Science and Technology*, John Wiley & Sons, Inc., Published Online: 15 JAN 2002. DOI: 10.1002/0471443395.img093., available online at <http://onlinelibrary.wiley.com/doi/10.1002/0471443395.img093/full>, last accessed on 08.02.2017.
- [52] B. G. Blundell, "Enhanced Visualization: Making Space for 3-D Images", Eds. John Wiley & Sons, 2007, ISBN: 978-0-471-78629-0
- [53] D. Lanman, M. Hirsch, Y. Kim, R. Raskar, "Content-Adaptive Parallax Barriers: Optimizing Dual-Layer 3D Displays using Low-Rank Light Field Factorization", in *ACM SIGGRAPH Asia 2010 papers (SIGGRAPH ASIA '10)*. ACM, New York, NY, USA , Article 163
- [54] SK. Kim, KY. Yoon, SK. Yoon, H. Ju, "Parallax barrier engineering for image quality improvement in an autostereoscopic 3D display", in *Optics Express*, vol. 23, issue 10, 05/2015, p. 13230
- [55] H. Xie, X. Zhao, Y. Yang, et al., "Cross-lenticular lens array for full parallax 3-D display with crosstalk reduction", in *Science China Technological Sciences*, March 2012, Volume 55, Issue 3, pp 735–742
- [56] H.J. Jeon, G. Park, J.S. Gwag, J.H. Lee, J.H. Kwon, "Color-Matching Liquid Crystal Display using a Lenticular Lens Array and RGB Light Sources", in *Journal of the Optical Society of Korea*. 2014. Aug, 18(4), pp. 345-349
- [57] D. Murph, "2010 Range Rover gets 12-inch 'dual view' touchscreen" , available online at <https://www.engadget.com/2009/04/08/2010-range-rover-gets-12-inch-dual-view-touchscreen/>, last accessed on 05.02.2017.
- [58] Burton Inc. homepage, available online at <http://burton-jp.com/en/index.htm>, last accessed on 05.02.2017.
- [59] Y. Ochiai, K. Kumagai, T. Hoshi, J. Rekimoto, S. Hasegawa, Y. Hayasaki, "Fairy Lights in Femtoseconds: Aerial and Volumetric Graphics Rendered by Focused Femtosecond Laser Combined with Computational Holographic Fields", in *ACM Transactions on Graphics*, Volume 35, Issue 2, May 2016, DOI: 10.1145/2850414. ,

available online at <https://arxiv.org/ftp/arxiv/papers/1506/1506.06668.pdf>, last accessed on 08.02.2017.

[60] M. Lucente, "Diffraction-Specific Fringe Computation for Electro-Holography", Doctoral Thesis Dissertation, MIT Dept. of Electrical Engineering and Computer Science, Sept. 1994.

[61] M. Harries, "Scare Quotes from Shakespeare: Marx, Keynes, and the Language of Reenchantment", Stanford University Press, 2000, pp. 23–41, ISBN 0-8047-3621-9.

[62] US8279268 B2 patent, "Projection system with wall structures for aerial display of three-dimensional video images", available online at <https://www.google.ch/patents/US8279268>, last accessed on 06.02.2017.

[63] Provision Holding, Inc. homepage, available online at <http://www.provision.tv/>, last accessed on 06.02.2017.

[64] Dreamoc homepage, available online at <http://www.dreamoc.nl/>, last accessed on 06.02.2017.

[65] D. Bernstein, "Making Something Out of Nothing", The New York Times, Dec. 18, 2003, available online at <http://www.nytimes.com/2003/12/18/technology/making-something-out-of-nothing.html>, last accessed on 06.02.2017.

[66] IO2 Technology homepage, available online at <http://www.io2technology.com/>, last accessed on 06.02.2017.

[67] Tactisplay table, available online at <http://www.tactisplay.com/product/tactisplay-table>, last accessed on 06.02.2017.

[68] B. Long, S. A. Seah, T. Carter, S. Subramanian, "Rendering Volumetric Haptic Shapes in Mid-Air using Ultrasound", Proceedings of ACM SIGGRAPH Asia, 2014 ACM Transactions on Graphics, 33(6),[181], DOI: 10.1145/2661229.2661257

[69] N. C. Bambakidis, W. R. Selman, A. E. Sloan, "Surgical rehearsal platform: potential uses in microsurgery", Neurosurgery, 2013 Oct; 73 Suppl 1:122-6, doi: 10.1227/NEU.0000000000000099.

[70] A. K. Leichman, "Better brain surgery, with flight simulation tech", Israel 21 C, Dec. 17, 2012, available online at <http://www.israel21c.org/better-brain-surgery-with-flight-simulation-tech/>, last accessed on 06.02.2017.

[71] Geomagic Touch homepage, available online at <http://www.geomagic.com/en/products/phantom-omni/overview>, last accessed on 06.02.2017.

[72] MedaPhor homepage, available online at <https://www.medaphor.com/>, last accessed on 06.02.2017.

- [73] I. Virag, L. Stoicu-Tivadar, E. Amaricai, "Browser-based medical visualization system," IEEE 9th International Symposium on Applied Computational Intelligence and Informatics (SACI, 2014), pp. 355-359, DOI: 10.1109/SACI.2014.6840092.
- [74] I. Virag, L. Stoicu-Tivadar, "A Survey of Web Based Medical Imaging Applications," Acta Electrotehnica, 2015, nr. 3, pp. 365 - 368, Editura Technical University of Cluj-Napoca, Romania
- [75] P. Nagy, "Open Source in Imaging Informatics," Journal of Digital Imaging, vol. 20, Suppl. 1, pp. 1-10, November 2007, DOI: 10.1007/s10278-007-9056-1.
- [76] T. Karopka, H. Schmuhl and H. Demski, "Free/Libre Open Source Software in Health Care: A Review," Journal: Healthcare Informatics Research, 2014, Volume 20, Number 1, Page 11, DOI: 10.4258/hir.2014.20.1.11.
- [77] T. Nakane, "Molecular Visualization, Online and On-the-Go", Research Collaboratory for Structural Bioinformatics Protein Data Bank newsletter, 2013, number 56, pp. 6, available online at [http://cdn.rcsb.org/rcsb-pdb/general\\_information/news\\_publications/newsletters/2012q4/education-corner.html](http://cdn.rcsb.org/rcsb-pdb/general_information/news_publications/newsletters/2012q4/education-corner.html), last accessed on 03.04.2017.
- [78] DICOM Transfer Syntax, available online at [ftp://dicom.nema.org/medical/DICOM/2013/output/chtml/part05/chapter\\_10.html](ftp://dicom.nema.org/medical/DICOM/2013/output/chtml/part05/chapter_10.html), last accessed on 03.04.2017.
- [79] I. Heikkinen, "Writing Augmented Reality Applications using JSARToolKit", HTML5 Rocks, 28.02.2012, available online at [https://www.html5rocks.com/en/tutorials/webgl/jsartoolkit\\_webrtc/](https://www.html5rocks.com/en/tutorials/webgl/jsartoolkit_webrtc/), last accessed on 03.04.2017.
- [80] Augmented Reality Toolkit homepage, available online at <http://www.hitl.washington.edu/artoolkit/>, last accessed on 03.04.2017.
- [81] NyARToolKit homepage, available at <http://nyatla.jp/nyartoolkit/wp/>, last accessed on 03.04.2017.
- [82] FLARToolKit homepage, available online at <http://www.libspark.org/wiki/saqoosha/FLARToolKit/en>, last accessed on 03.04.2017.
- [83] Cornerstone Core homepage, available online at <https://github.com/chafey/cornerstone>, last accessed on 04.04.2017.
- [84] cornerstonWADOImageLoader homepage, available online at <https://github.com/chafey/cornerstoneWADOImageLoader>, last accessed on 04.04.2017.
- [85] DICOM Web Access to DICOM Objects standard, available online at <http://www.dclunie.com/dicom-status/status.html#Supplement85>, last accessed on 04.04.2017.

- [86] dicomParser homepage, available online at <https://github.com/chafey/dicomParser>, last accessed on 04.04.2017.
- [87] cornerstoneTools homepage, available online at <https://github.com/chafey/cornerstoneTools>, last accessed on 04.04.2017.
- [88] HTML 5 W3C Recommendation homepage, available online at <https://www.w3.org/TR/2014/REC-html5-20141028/>, last accessed on 04.04.2017.
- [89] HTML 5.1 W3C Recommendation homepage, available online at <https://www.w3.org/TR/2016/REC-html51-20161101/>, last accessed on 04.04.2017.
- [90] HTML5 Differences from HTML4, W3C Working Group Note 9<sup>th</sup> of December 2014, available online at <https://www.w3.org/TR/html5-diff/>, last accessed on 05.04.2017.
- [91] "WebRTC 1.0: Real-time Communication Between Browsers", W3C Working Draft, 13th of March 2017, available online at <https://www.w3.org/TR/webrtc/>, last accessed on 06.04.2017.
- [92] Rtcweb Status Pages, Real-Time Communication in WEB-browsers (Active WG), available online at <https://tools.ietf.org/wg/rtcweb/>, last accessed on 06.04.2017.
- [93] OpenWebRTC homepage, available online at <https://www.openwebrtc.org/>, last accessed on 06.04.2017.
- [94] vLine - Cloud Infrastructure for WebRTC, available online at <https://github.com/vline>, last accessed on 06.04.2017.
- [95] Sharefest homepage, available online at <https://www.sharefest.me/>, last accessed on 06.04.2017.
- [96] WebGL homepage, available online at <https://www.khronos.org/webgl/>, last accessed on 06.04.2017.
- [97] ThreeJS homepage, available online at <https://threejs.org/>, last accessed on 06.04.2017.
- [98] I. Virag, L. Stoicu-Tivadar, M. Crisan-Vida, E. Amaricai, "Server-side Image Segmentation and Patient Related Data Storage," Proceedings of the 6th International Workshop on Soft Computing Applications (SOFA, 2014), pp. 259-266, DOI: 10.1007/978-3-319-18296-4\_22
- [99] I. Virag, L. Stoicu-Tivadar, M. Crisan-Vida, "Gesture-Based Interaction in Medical Interfaces," IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI, 2016), pp. 519 - 523, DOI: 10.1109/SACI.2016.7507339

- [100] I. Virag, L. Stoicu-Tivadar, M. Crisan-Vida, "Client-side Medical Image Colorization in a Collaborative Environment," *Studies in Health Technology and Informatics*, vol. 210: Digital Healthcare Empowering Europeans, R. Cornet et al. (Eds.), IOS Press, May 2015, pp. 904 – 908, DOI 10.3233/978-1-61499-512-8-904
- [101] M. Martinez Escobar, "An interactive color pre-processing method to improve tumor segmentation in digital medical images," master's thesis, Iowa State University, 2008, available online at <http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=2096&context=etd>, last accessed on 09.04.2017.
- [102] OsiriX homepage, available online at <http://www.osirix-viewer.com/>, last accessed on 09.04.2017.
- [103] VolView homepage, available online at <http://www.kitware.com/opensource/volview.html>, last accessed on 09.04.2017.
- [104] I. Virag, L. Stoicu-Tivadar, M. Crisan-Vida, "Gesture Interaction Browser-Based 3D Molecular Viewer," in *Unifying the Applications and Foundations of Biomedical and Health Informatics*, J. Mantas et al. (Eds.), series of *Studies in Health Technology and Informatics*, vol. 226, IOS Press, 2016, pp. 17 - 20, DOI 10.3233/978-1-61499-664-4-17
- [105] J. Rubin, D. Chisnell, "Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd Edition," Wiley Publishing Inc., 2008, ISBN: 978-0-470-18548-3
- [106] Usability Test Plan Template, available online at <https://www.usability.gov/how-to-and-tools/resources/templates/usability-test-plan-template.html>, last accessed on 10.04.2017
- [107] Consent Form (Adult) Template, available online at <https://www.usability.gov/how-to-and-tools/resources/templates/consent-form-adult.html>, last accessed on 10.04.2017
- [108] J. Kirakowski, M. Corbett, "SUMI - The Software Usability Measurement Inventory," *British Journal of Educational Technology*, (1993) Volume 24, Issue 3, pp. 210-212, DOI: 10.1111/j.1467-8535.1993.tb00076.x
- [109] ISO 9241-9:2000, "Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 9: Requirements for non-keyboard input devices," available online at <https://www.iso.org/standard/30030.html>, last accessed on 11.04.2017
- [110] I. Scott MacKenzie, "What is ISO 9241-9," presentation, available online at <https://pdfs.semanticscholar.org/f13f/1e47f528759fd992336ca2b08cca3072268f.pdf>, last accessed on 11.04.2017